

WEBSTER COMPUTER CORPORATION  
WQESD Winchester Disc Controller

USER MANUAL

Version 6.3  
October 1990

Document Number : 1583801

WEBSTER COMPUTER CORPORATION

USA

2109 O'Toole Avenue, Suite J  
San Jose, CA 95131-1303

USA

Voice (408) 954 8054

FAX (408) 954 1832

AUSTRALIA

1270 Ferntree Gully Road  
Scoresby, Victoria 3179

AUSTRALIA

Voice (03) 764 1100

Service (03) 764 1156

FAX (03) 764 1179

First Floor 9-19 Ryde Road  
Pymble, New South Wales 2073

AUSTRALIA

Voice (02) 498 4422

FAX (02) 499 2147

163 Wharf Street  
Spring Hill, Queensland 4000

AUSTRALIA

Voice (07) 832 1899

FAX (07) 832 2302

UNITED KINGDOM

Unit 7 Weltech Centre  
Ridgeway  
Welwyn Garden City  
Hertsfordshire AL72AA

UNITED KINGDOM

Voice (707) 33 6969

FAX (707) 37 3378

## TABLE OF CONTENTS

INSTALLATION PROCEDURE . . . . .	1
WQESD CONTROLLER SPECIFICATIONS . . . . .	2
CHAPTER 1    WQESD GENERAL DESCRIPTION . . . . .	3
CHAPTER 2    WQESD INSTALLATION . . . . .	8
2.1    WQESD Switch Settings . . . . .	8
Base (CSR) Address . . . . .	8
Automatic Bootstrap Select . . . . .	8
Power Up Mode . . . . .	9
Interrupt Priority . . . . .	10
2.2    KESDI/T RS232 Maintenance Terminal Adaptor	11
2.3    Connecting Multiple Drives . . . . .	12
2.4    Front Panel Connections . . . . .	12
2.5    On-Board Bootstrap . . . . .	15
CHAPTER 3    WQESD WOMBAT UTILITIES . . . . .	18
3.1    Starting Up WOMBAT . . . . .	18
LSI-11 CPU . . . . .	20
MicroVAX II . . . . .	21
3.2    WOMBAT Menu Options . . . . .	21
3.3    Master Menu Options . . . . .	22
3.4    Disc Structure Menu Options . . . . .	24
3.5    Disc Test Menu . . . . .	30
3.6    Bad Block Management Menu . . . . .	32
3.7    Shadow Options Menu Option . . . . .	35
3.8    WOMBAT Disc Structure . . . . .	38
3.9    Drives with Removable Media . . . . .	39
3.10   Error Recovery Procedures . . . . .	39
3.11   Drive Shadowing . . . . .	39
3.12   Computing Sectors Per Track . . . . .	40
3.13   WOMBAT Error Messages . . . . .	41
3.14   WOMBAT Self Diagnostics . . . . .	43
CHAPTER 4    WQESD MSCP PROGRAMMING . . . . .	44

4.1	Overview of MSCP . . . . .	44
4.2	Controller Communications . . . . .	44
4.3	Message Transmission . . . . .	49
4.4	Data Transmission . . . . .	50
4.5	Initialization . . . . .	50
4.6	Registers . . . . .	51
4.7	MSCP Commands . . . . .	53
4.8	Error Handling . . . . .	54
4.9	Fatal Controller Error . . . . .	54
CHAPTER 5	WQESD OPERATING SYSTEMS . . . . .	56
5.1	Operating Systems Overview . . . . .	56
5.2	RT-11 Operating System . . . . .	58
5.3	RSTS/E Operating Systems . . . . .	62
5.4	RSX-11M Operating System . . . . .	63
5.5	RSX-11M-PLUS Operating System . . . . .	66
5.6	MicroVAX/MicroVMS Operating System . . . . .	70
5.7	Autoconfigure . . . . .	72
5.8	Error Logging . . . . .	79
CHAPTER 6	WQESD CACHE OPERATION . . . . .	82
6.1	WQESD Disc Cache . . . . .	82
6.2	Read Look-ahead . . . . .	82
6.3	Cache Allocation . . . . .	82
6.4	Cache Usage . . . . .	83
6.5	Cache Assignment Algorithm . . . . .	83
6.6	Cache Operation . . . . .	83
6.7	Cache Disable . . . . .	84
6.8	Early Write Notification . . . . .	84
CHAPTER 7	WQESD QBUS INTERFACE . . . . .	85
7.1	Qbus Interface . . . . .	85
7.2	Interrupts . . . . .	85
7.3	Direct Memory Access . . . . .	86
CHAPTER 8	WQESD ESDI INTERFACE . . . . .	88
8.1	ESDI Interface . . . . .	88
8.2	Control and Data Cables . . . . .	88
8.2	Command Data . . . . .	89
CHAPTER 9	WQESD HARDWARE DESCRIPTION . . . . .	92
9.1	General Description . . . . .	92
9.2	Detailed Description . . . . .	93
	Microprogrammed Sequencer . . . . .	93
	Qbus Interface . . . . .	94

	ESDI Interface . . . . .	95
	Cache Memory . . . . .	96
	I/O Port . . . . .	96
CHAPTER 10	WQESD CIRCUIT DIAGRAMS . . . . .	98
APPENDIX A	SAMPLE SWITCH SETTINGS . . . . .	106

## TABLES

2-1	BASE (CSR) ADDRESS - SWITCH (N6) .....	8
2-2	AUTOMATIC BOOTSTRAP SELECT - SWITCH (N6) .....	9
2-3	POWER UP MODE - SWITCH (W3) .....	9
2-4	INTERRUPT PRIORITY - SWITCHES (X8) .....	10
2-5	KESDI/T RS232 MAINTENANCE TERMINAL ADAPTOR .....	11
2-6	10-WAY FRONT PANEL/MAINTENANCE CONNECTOR .....	13
2-7	CSR ADDRESS FOR WQESD BOOTSTRAP .....	17
3-1	WOMBAT INITIALISATION CODES .....	19
3-2	WQESD CSR ADDRESSES .....	19
4-1	MSCP COMMAND RING CODE DEFINITIONS .....	46
4-2	MSCP WORD ENVELOPE CONTENTS .....	48
4-3	MSCP INITIALISATION PARAMETERS .....	51
4-4	WQESD INITIALISATION WORDS .....	52
4-5	WQESD MSCP COMMANDS .....	53
4-6	FATAL CONTROLLER ERRORS .....	54
4-7	MSCP STATUS CODE MESSAGES .....	55
5-1	DEVICE NAMES IN DEC OPERATING SYSTEMS .....	57
5-2	SYSGEN DEVICE RANKING .....	73
5-3	DEVICE REGISTERS AND WORD BOUNDARIES .....	74
5-4	FLOATING VECTOR ADDRESS DEVICE PRIORITY RANKING .....	75/76
5-5	CSR AND VECTOR ADDRESS EXAMPLE .....	77
5-6	FLOATING CSR ADDRESS ASSIGNMENT .....	78
8-1	ESDI CONTROL CABLE (J1/P1) SIGNALS .....	88
8-2	ESDI INTERFACE DATA CABLE (J2/P2) SIGNALS .....	89
8-3	ESDI COMMAND DATA WORD STRUCTURE .....	90
8-4	ESDI COMMAND (CMD) DATA DEFINITION .....	91

## FIGURES

2-1	WQESD SWITCH FACTORY SETTINGS .....	14
4-1	MSCP MEMORY COMMUNICATIONS FORMAT .....	45
4-2	MSCP DESCRIPTOR FORMAT .....	45
4-3	MSCP MESSAGE ENVELOPE FORMAT .....	47

## Installation Procedure

The installation of the WQESD is relatively simple. Before the WQESD can be used the following procedure must be followed.

1. If you are not familiar with the WQESD and its features, **READ this manual's introductory chapter (Chapter 1).**
2. Select the controller CSR address and interrupt priority by setting the switches located on the controller. See Chapter 2 - Installation.
3. Locate the controller in the Qbus backplane and attach the drives. The drive data cables are connected in radial fashion while the control cable is daisy chained with the last drive being terminated.
4. Power-up the system and invoke WOMBAT using one of the methods outlined in Chapter 3 - WOMBAT Utilities. WOMBAT is Webster's interactive diagnostic and formatting program. **The disc(s) cannot be used with the WQESD until WOMBAT has structured and formatted the disc(s).** Answer the WOMBAT prompts according to the disc structure and system configuration you have selected.

When all these steps have been completed the WQESD will be ready to use with your computer system.

## WQESD CONTROLLER SPECIFICATIONS

Bus interface	DEC Qbus
Transfer mode	Block mode DMA
Memory address capacity	4 Megabyte (22 bit)
Software Emulation	DEC MSCP
Command buffer capacity	Up to 32 commands
Disc Cache Size	1 Megabyte (with parity)
Qbus loads	1 DC, 1 AC
Drive interface	ESDI (Enhanced Small Device Interface) For both Hard and Soft sectorized drives
Max. No. of drives	4
Max. No. of cylinders	4096
Max. No. of heads	16 groups of 16 heads
Bytes per Sector	512
Max. Sectors per Track	255
Max Disc size	18 Gigabytes
Access time overhead	4 ms (plus drive access time)
Transfer rate	3.0 Mbyte/sec maximum non interleaved
Drive Connectors	34 way control 4 x 20 way data
Power requirement	5V 3.5A typical
Physical Size	226mm x 266mm (QBus quad)
Temperature	5 - 60 degrees C. operating 0 - 66 degrees C. store
On board LED indicators	Red: Fatal error indicator Green: Access in progress
TTL outputs	Disc or cache access in progress
RS232 output	Data transmitted to terminal - 9600 baud
RS232 input	Data received from terminal- 9600 baud

### Optional Accesories:

KESDI/1,/2,/3,/4	Drive and control cable kits configured for one to four drives
KESDI/T	RS232 Maintenance Terminal Adaptor kit



## Chapter 1 General Description

### Overview

The Webster WQESD is a high performance quad height interface to ESDI compatible disc drives. The WQESD can support up to four ESDI drives and features a one megabyte cache memory, 24 megabit per second throughput, command queueing, overlapped seeks, and implements DEC's Mass Storage Control Protocol (MSCP).

The Webster WQESD flexibly couples discs of any size and data rate to all standard DEC operating systems without software modification. Comprehensive on-board interactive formatting and diagnostic firmware provides engineering support across the range of LSI-11, MicroVAX and various non DEC implementations of the Qbus.

### The ESDI Interface

The ESDI or Enhanced Small Device Interface is a low cost, high performance interface suitable for smaller high performance hard disc drives.

### One Megabyte Disc Cache

All data is read from and written into the cache. Data transfers from the cache are approximately 3 ms compared to 30 - 38 ms for typical drive access times - up to an 93 percent reduction in access time. The cache memory acts as a track buffer and can provide data at rates of 3 megabytes per second.

### Four ESDI drives at 2.4 Megabytes per second

The Webster WQESD controller will support up to four ESDI interface drives. All four drives can have different storage capacities, speed and data rates. This allows the user to match drive characteristics with applications.

### No Sector Interleaving

The WQESD stages all data from the disc drive through the cache memory. This ensures that all data can be transferred at full disc speed over the disc interface and at maximum speed over the Qbus without incurring data late errors. Disc and Qbus transfers are performed simultaneously to minimize access times.

### Read Look-ahead

The Webster WQESD allows the user to program the controller to perform 'look-ahead' reads in anticipation of data requests. Whole tracks or more can be read into the cache ensuring that data is ready for the host computer the instant its needed.

### Virtual Units

The Webster WQESD also allows the user to partition each drive into virtual units which are addressed by the host as individual drives. Each virtual unit can be any size up to the size of the entire drive with up to 16 virtual units assigned to each controller. Each virtual unit can be further partitioned under the host operating system.

### Block mode DMA and DMA Throttle

With Block Mode DMA, the Webster WQESD interleaves address references with bursts of data - almost doubling Qbus throughput. The WQESD fully conforms with Qbus Block Mode DMA protocol. With non block mode memory, the WQESD automatically reverts to burst mode DMA.

After every 16 word DMA transfer there is a 4 microsecond delay to service any pending interrupt or DMA requests from other devices. If a DMA request occurs a 'DMA throttle' will release the Qbus after 8 words to prevent data loss from other DMA devices.

### Drive Shadowing

The Webster WQESD offers the user the option of drive shadowing. Data integrity is further improved by writing the same data to two drives simultaneously. In certain circumstances greater data throughput may be achieved because the controller can decide which drive has its heads positioned closest to the required data, and schedule a read on that drive.

### Seek Optimisation and Overlap

The Webster WQESD can queue up to 32 commands. The optimum order of execution is dynamically computed according to the strategy selected by the user. With multiple disc drives seeks are initiated simultaneously, further improving performance.

## Error Checking and Correction

The Webster WQESD uses a 48-bit ECC polynomial with an 11-bit correction span for error detection and correction. The WQESD will try up to 10 times to correct an error before reporting the fault to the host system.

## Dynamic Bad Block Replacement

The Webster WQESD dynamic bad block replacement and error correction always presents error free 'perfect media' to the host computer. During normal operation the controller dynamically replaces any blocks it detects as bad with an alternative block from a replacement block pool. Blocks with hard errors are replaced but the data in them flagged 'forced error'. This indicates to the host that though the data in these blocks is bad the blocks themselves are now good. All bad block replacement is completely invisible to the host computer.

## Statistics Recording

The Webster WQESD records statistics such as the number of reads and writes, cache hits and misses, and other important information for each drive. The user can interrogate the drive for this information according to application specific performance requirements.

## Write Protect

A connector is provided to which the user can connect one write protect switch per drive.

## MSCP Emulation

The Webster WQESD communicates with the host through a simple register pair to memory resident 'command packets'. Disc geometry factors such as sectors, heads, cylinders and disc capacity are invisible to the host computer. The Webster WQESD accepts 32 bit binary block numbers and converts them to physical disc addresses, allowing any size disc to be fully accessed by any program without software modification.

Supported operating systems include RT-11 version 5; RSX-11M-Plus version 3; TSX-Plus version 6; RSTS/E version 9; MicroVMS version 5; or later versions. Various UNIX versions are also supported.

## **Qbus interface**

Originally introduced in 1975 by Digital Equipment Corporation to support the LSI-11 CPU range, the Qbus architecture has evolved in speed and function to the point where it now outperforms most small computer bus systems. The WQESD fully implements all current Qbus enhancements, including block mode transfers, 4-level interrupt structure, 22 bit addressing, and fully supports LSI11/2, LSI11/23, LSI11/73, MicroVAX II and MicroVAX 3000 Qbus CPU designs.

## **On-Board WOMBAT Utilities**

Webster Omnipotent Mass Builder and Tester (WOMBAT) is an interactive formatting and diagnostic utility contained within the WQESD firmware. An on-board serial connection allows WOMBAT to be run using an ASCII terminal. This permits disc formatting and maintenance operations to be carried out with minimal additional hardware.

WOMBAT can also load a simple console communication program into the host computer's memory or it can be invoked on system power-up. No external software, media, or program loading device is required in maintenance of the WQESD or its attached disc drives. WOMBAT is always available independently of the host CPU type or the operating system environment.

## **WOMBAT Formatter**

WOMBAT initialises a fresh disc drive by writing sector addresses and zero data blocks through the entire recording surface. WOMBAT prompts the user at the terminal to supply parameters such as drive geometry (cylinders, heads and sectors) and various other options. This data is stored twice in special reserved areas of track zero and retrieved by a simple homeseek-read sequence at each power-up. No special PROMs or switch settings are required to fully characterize the connected disc drives.

## **WOMBAT Self Diagnostics**

The WQESD contains a comprehensive set of self diagnostic procedures which are executed automatically on power-up. Failure is indicated by a flashing red LED and a fatal error status which is deposited in the SA register.

### **WOMBAT Interactive Diagnostics**

Terminal oriented engineering utilities contained within the WOMBAT firmware include a continuous read/write/seek exerciser, a disc surface pattern tester and a bad block replacement routine.

### **Sequential Spin-up**

The WQESD controller will spin-up ESDI drives in a sequential manner, providing they have the "motor control" option enabled. This is done to minimise start-up current surge.

## Chapter 2 WQESD Installation

### 2.1 WQESD Switch Settings

The WQESD can be configured by selecting the appropriate switch combination. The switch positions for base (CSR) address, automatic bootstrap select, interrupt priority, and power up mode are given below. Note that there is no interrupt vector switch as this is set automatically by the operating system software.

#### BASE CSR ADDRESS

The base CSR address should be selected according to the rules set out in Chapter 5 - Operating Systems. It is important to locate the controller correctly to enable the operating system to identify the device and its type. The following is a table of all the possible CSR addresses for LSI-11 and MicroVAX systems.

TABLE 2-1 BASE (CSR) ADDRESS - SWITCH (N6)

ADDRESS		LSI-11	MicroVAX II
1st (A1)	SW10 -ON	17772150	20001468
2nd (A2)	SW9 - ON	17760334	200000DC
3rd (A3)	SW8 - ON	17760354	200000EC
4th (A4)	SW7 - ON	17760374	200000FC
5th (A5)	SW6 - ON	17760414	2000010C
6th (A6)	SW5 - ON	17760434	2000011C

NOTE - Only one of the above six switches should be on at any one time.

#### AUTOMATIC BOOTSTRAP

The on-board bootstrap is provided to allow systems to boot from the controller. The switch settings for the bootstrap are given below. The bootstrap procedure is given in section 2.5. The bootstrap must be disabled before being installed in a MicroVAX.

TABLE 2-2     AUTOMATIC BOOTSTRAP SELECT - SWITCH (N6)

SWITCH	FUNCTION
B1,SW3 - ON	On board bootstrap disabled
B2,SW2 - ON	On board bootstrap enabled at address 17773000
B3,SW1 - ON	On board bootstrap enabled at address 17771000

NOTE - Only one of the above three switches should be on at any one time. Switch 4 is unused.

#### POWER UP MODE

This feature is used in combination with the automatic bootstrap. If the bootstrap is enabled the appropriate power up mode must be selected. Switch 3 is set "OFF" to run WOMBAT when the system is booted. This feature does not apply to the MicroVAX II.

TABLE 2-3     POWER UP MODE - SWITCH (W3)

SWITCH NUMBER	SETTING	FUNCTION AT POWER UP
SW2 (S1)	ON	Not DP8466 Rev G
	OFF	DP8466 Rev G
SW3 (S0)	ON	Normal Power Up
	OFF	Start WOMBAT on Power Up
SW4	ON	Disable Sequential Drive Spinup
	OFF	Enable Sequential Drive Spinup

Switch S2 selects the revision of Disk Controller IC used on the WQESD. This switch is set correctly at the factory, but its setting may have changed over time. If you are configuring a new WQESD board, this switch does not need to be changed.

If you are configuring an old WQESD board, or wish to check whether the switch is set correctly, use the guidelines below to determine

the revision of Disk Controller IC installed. The Disk Controller IC (part number DP8466) is a large 48 pin IC which occupies location F3 (adjacent to the cache array).

If the Disk Controller IC is labelled "DP8466AN-12" or "DP8466AN-20" it is a "Rev G" DP8466, and Switch 2 (S1) should be in the "OFF" position.

Any Disk Controller IC that does NOT have an "A" immediately after the 8466 in the IC's part number is NOT a "Rev G" DP8466. Switch 2 (S1) should be in the "ON" position.

Sequential drive spinup, enabled by setting Switch 4 "OFF", causes drives connected to the WQESD to be powered up in turn. This can be useful in drive sub-systems with limited power supply capacity, as it reduces the load on the power supply by not allowing all drives to draw their startup current simultaneously.

With Switch 4 "ON", all drives connected to the WQESD will spin up when power is applied to the drives.

Note: Switch S1 is not used.

#### INTERRUPT PRIORITY SETTING

Interrupts suspend program execution while the processor starts the device service routine at a vector address input from the requesting device.

Interrupts are serviced according to device priority. Device priority can be determined in two ways. These are termed 'Position Defined' and 'Distributed' arbitration. Positioned Defined arbitration gives priority to those devices which are electrically closest to the processor. Distributed arbitration implements priority according to the priority levels set on the device hardware. When devices with equal priority generate an interrupt, the processor gives preference to the device which is electrically closest. A previous bus transaction must have been completed before another can be commenced.



TABLE 2-4 INTERRUPT PRIORITY - SWITCH (X8)

(P3) SW2,	(P2) SW3,	(P1) SW4	PRIORITY
ON	ON	ON	4
ON	ON	OFF	5
ON	OFF	ON	6
OFF	OFF	ON	7

NOTE - Switch one is unused.

## 2.2 KESDI/T RS232 Maintenance Terminal Adaptor

The KESDI/T RS232 maintenance terminal adaptor allows the controller to be connected to an ASCII terminal. It consists of a 10-way flat cable with a DB25S connector on one end and a 10-way insulation displacement type flat cable socket on the other.

The communication format is :

ASCII RS232 9600 Baud, 7 Data Bits, 1 Stop Bit, no parity.

Note that if a VT220 terminal is used the communication format must be set-up for space parity rather than no parity.

If normal disc access is attempted with this cable connected to a terminal, garbage will appear on the terminal due to the shared RS232 Output/Access Light Function. This is normal.

TABLE 2-5 KESDI/T RS232 MAINTENANCE TERMINAL ADAPTOR

J6 Pins	DB25S Pins	Function
7	7	RS232 Enable
8	2	RS232 Input
3	3	RS232 Output
4	7	Ground

## 2.3 Connecting Multiple Drives

The WQESD supports up to four ESDI drives each of which can be of different size, speed and data rate. The procedure for connecting multiple drives is simple. Data cables are connected in radial fashion to edge connectors on the WQESD. There is no particular order of connection. The control cable is daisy chained between each drive with the last drive being terminated.

Pre-configured data and control cable kits for one to four drives can be supplied to minimise difficulties when installing the WQESD or where adding additional drives. (KESDI/1, KESDI/2, KESDI/3, KESDI/4)

Cables should be checked for correct installation by ensuring that Pin 1 on the edge connector aligns with Pin 1 on the cable. Pin 1 is generally indicated by the arrow mark on the edge connector shroud. If there is no shroud Pin 1 is the top right pin with the edge connectors facing you. Pin 1 on the cable is usually indicated by a coloured stripe on the appropriate cable.

Correct drive termination is also necessary. The disc drive technical literature should be consulted for the appropriate termination procedure.

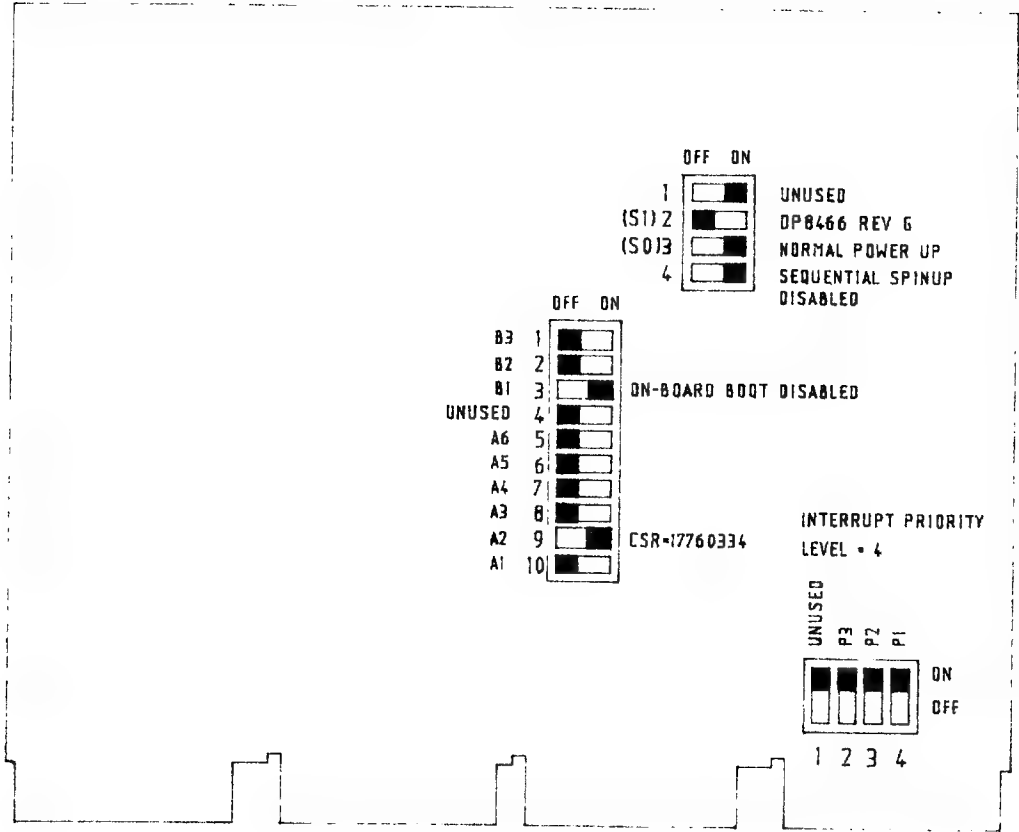
## 2.4 Front Panel Connections

If required, a front panel can be connected to J6, the front panel/maintenance connector. There are three possible options for the front panel - an active panel, a passive panel or no panel. The active front panel provides write protection for all drives, an online/offline button and a drive ready indicator. The passive panel provides write protection and access light for two drives only. (See Table 2-6). The active front panel must contain some digital circuitry and be separately powered to support these functions. Webster Computer Corporation can provide full details on request. Note that the correct panel-type must be selected in the WOMBAT Disc Structure Menu (Section 3.4).

TABLE 2-6 PASSIVE 10-WAY FRONT PANEL/MAINTENANCE CONNECTOR

Pin No.	Front Panel Function	Maintenance Function
7	Drive 1 Write Protect Input Connecting this input to Ground will write-protect the drive. Has an on-board 180 ohm pull-up.	RS232 Enable
8	Drive 0 Write Protect Input Connecting this input to Ground will write-protect the drive. Has an on-board 22k ohm pull-up.	RS232 Input
1	Drive 1 Online/Offline Switch Input Connecting this input to Ground will prevent access to the drive. Has an on-board 180 ohm pull-up.	
10	Drive 0 Online/Offline Switch Input Connecting this input to Ground will prevent access to the drive. Has an on-board 180 ohm pull-up.	
3	Access Light. Indicating access to RS232 Output Data Drive 0, Drive 1 or Cache. This output can be used to drive an access light. The levels are : ACCESS -5 v through 1.5k ohm NO ACCESS : +4.5 @ 20mA max.	
5	TTL signal indicating access to Drive 0. Low true.	
6	TTL signal indicating access to Drive 1. Low true.	
2,4,9	Ground	Ground

FIGURE 2-1 WQESD FACTORY SWITCH SETTINGS



## 2.5 On-Board Bootstrap (LSI-11 only)

If the WQESD bootstrap is enabled the following occurs :-

1. On initialisation location 773000 responds to the CPU fetch instruction with a BR .+2 (400) instruction to address 773002 where the CPU loops on a BR . instruction (777).
2. A "jump to zero" instruction (JMP @#0 (137, 000)) is loaded into locations zero and two.
3. The controller program forces the CPU to start executing at location zero by changing the contents of location 773002 to CLR PC (5007).
4. Bootstrap code is loaded into host memory at location 2000 and the rest of memory is cleared.
5. The controller changes location 2 from zero to 2000 changing the JMP @#0 to JMP @#2000, starting the execution of the loaded bootstrap program and the following message is typed :-.

### BOOT V2.0.

6. The controller boot program allows approximately 2 seconds for the operator to strike any key on the keyboard. If no key is struck the boot types :-

#### Booting from DU0:

The bootstrap program reads in that device's boot block starting at location zero.

7. If any key is struck by the operator within two seconds the boot prompts with :-

>

The operator may then key in a device DU, DL, DY, MS or W. Note that device **W** will invoke WOMBAT. By further specifying **A, B, C, D, E** or **F** after **W**, WOMBAT on controller **A, B, C, D, E** or **F** will be invoked. For example, **WB** will invoke WOMBAT on controller **B**.

The following is the syntax of the WQESD bootstrap procedure :-

> [ DEVICE ] -- [ Controller Number ] -- [ Unit Number ] --[:]

such that :-

```

      ^-->DU - [A,B,C,D,E,F]-[0-7] ---- [:]
      |
      |-->DL ----- [0-7] --|
      |
      |-->DY ----- [0-1] --|
      |
      |-->W ----- [A,B,C] -----|
      |
      |-->MS --- [A,B,C,D] -----

```

8. If any error occurs, a message from the following set is printed and the boot re-prompts for step five.

"? - Device must be DU, DY,DL or MS"

"Unit must be 0 - 7" - For DU or DL

"Unit must be 0 or 1" - For DY

"Controller must be A, B or C" - For DL

"Controller must be A, B, C or D" - For MS

"Boot failure" - Device unavailable (or not ready if DL)

9. The selected controller is commanded to read the boot block (block zero) from the specified device into the host memory and then waits to be initialised. The host computer commences execution of the instructions in the boot block. The devices "assumed" by the WQESD bootstrap are detailed in Table 2-7.

TABLE 2-7 CSR ADDRESSES FOR WQESD BOOTSTRAP

DEVICE	ADDRESS
CONSOLE	177560
DU:	172150
DUA:	172150
DUB:	160334
DUC:	160354
DUD:	160374
DUE:	160414
DUF:	160434
MS:	172520
MSA:	172520
MSB:	172524
MSC:	172530
MSD:	172534
MU:	174500
MUA:	174500
DL:	174400
DY:	177170

NOTE: When starting WOMBAT the letter "W" may be followed by any letter available for "DU".

## Chapter 3 WQESD WOMBAT Utilities

### 3.1 Starting Up WOMBAT

WOMBAT (Webster Omnipotent Mass Builder and Tester) provides a controller resident means of formatting, testing and maintaining the drive and controller subsystem. All WOMBAT functions are menu driven and are designed to simplify the process of structuring, formatting and testing drives.

WOMBAT can be invoked using any of the following four methods :-

- (a) By selecting the 'W' option during the system bootstrap operation if the WQESD boot is enabled. The letter "W" may be followed by any letter allowed for "DU" to use any of the available CSRs.
- (b) By connecting a 9600 baud terminal to the WQESD Front Panel connection and depositing the appropriate data pattern using ODT.
- (c) By connecting a 9600 baud terminal to the WQESD and configuring the switches on the PCB to automatically run WOMBAT.
- (d) By depositing a special data pattern in the WQESD using Micro ODT from the system console.

Selecting option (a) or (d) allows disc testing and diagnostics to be undertaken from the user console. The console link is formed by a communication program which WOMBAT downloads into main memory.

WOMBAT can be run independently of, or without a CPU for controller testing or engineering purposes if necessary. First disable the controller bootstrap by setting switch B1, (SW3) to ON and the Power Up Mode (SW3) to OFF. (See Table 2-3). Connect a 9600 baud terminal to the Front Panel connector and position the controller in the backplane. WOMBAT will be invoked automatically on power-up or by pressing reset. If no CPU is present then the backplane must be correctly terminated and a bus intialisation signal (BINIT) must be generated.

To resume normal operation the configuration switches must be reset as required. Note that setting the Power Up Mode to "OFF" will make the controller completely unavailable to the host CPU.

Table 3-1 lists the octal codes used to invoke WOMBAT using console ODT.



TABLE 3-1 WOMBAT INITIALISATION CODES

	NUMBER		FUNCTION	PROCESSOR
	octal	hex		
	000250	00A8	WOMBAT	LSI-11
	000254	00AC	WOMBAT	MicroVAX II
	000260	00B0	WOMBAT	On board port

The procedures for invoking WOMBAT on LSI-11, and the MicroVAX II are given below. WOMBAT can be stopped by simply re-booting the system.

In each case input the appropriate CSR address for the controller to be accessed where indicated by 'CSR'. Table 3-2 lists the possible CSR addresses for the WQESD.

TABLE 3-2 WQESD CSR ADDRESSES

ADDRESS		LSI-11	MicroVAX II
<u>WCSR0</u>			
1st (A1)	SW10 - ON	172150	20001468
2nd (A2)	SW9 - ON	160334	200000DC
3rd (A3)	SW8 - ON	160354	200000EC
4th (A4)	SW7 - ON	160374	200000FC
5th (A5)	SW6 - ON	160414	2000010C
6th (A6)	SW5 - ON	160434	2000011C
<u>WCSR1</u>			
1st (A1)	SW10 - ON	172150	20001468
2nd (A2)	SW9 - ON	160334	200000DC
3rd (A3)	SW8 - ON	160360	200000F0
4th (A4)	SW7 - ON	160364	200000F4
5th (A5)	SW6 - ON	160370	200000F8
6th (A6)	SW5 - ON	160374	200000FC

.....cont.

TABLE 3-2 WQESD CSR ADDRESSES

ADDRESS	LSI-11	MicroVAX II
<u>WCSR2</u>		
1st (A1) SW10 - ON	172150	20001468
2nd (A2) SW9 - ON	166334	20000CDC
3rd (A3) SW8 - ON	166354	20000CEC
4th (A4) SW7 - ON	166374	20000CFC
5th (A5) SW6 - ON	166414	20000D0C
6th (A6) SW5 - ON	166434	20000D1C
<u>WCSR3</u>		
1st (A1) SW10 - ON	172150	20001468
2nd (A2) SW9 - ON	160334	200000DC
3rd (A3) SW8 - ON	160340	200000E0
4th (A4) SW7 - ON	160344	200000E4
5th (A5) SW6 - ON	160350	200000E8
6th (A6) SW5 - ON	160354	200000EC
<u>WCSR4</u>		
1st (A1) SW10 - ON	172150	20001468
2nd (A2) SW9 - ON	172154	2000146C
3rd (A3) SW8 - ON	172160	20001470
4th (A4) SW7 - ON	172164	20001474
5th (A5) SW6 - ON	172170	20001478
6th (A6) SW5 - ON	172174	2000147C

NOTE: The standard CSR options come with the WCSR0 PROM. Other CSRs can be obtained by ordering any of the optional address PROMs.

#### LSI-11 CPU

The following details the procedure for invoking WOMBAT on an LSI-11 CPU system using console ODT.

Halt the processor.

'CSR' /	000000	250	(ask WOMBAT to load the communications program)
R7 /	XXXXXX	2000	(set up the program start address)
RS /	000000	340	(set PSW to block interrupts)
P			(now start the program without a bus reset.)

## MicroVAX II

The following details the procedure for invoking WOMBAT in a MicroVax II using ODT.

Halt the CPU at the end of its start-up diagnostics by turning on the "halt enable" switch at the back of the CPU.

When it halts :-

D/P/W	20001F40	20	(enable QBUS access to memory)
D/L	20088008	80000002	(set-up the appropriate QBUS map entry)
D/W	'CSR'	AC	(ask WOMBAT to load the communications program into memory)
S	400		(start the program)

## 3.2 WOMBAT Menu Options

When WOMBAT is invoked it will display an announcement and then print a list of all drives and units and prompts for the drive number on which to perform operations.

WQESD WOMBAT Version: 2.0

UNIT	DRIVE	OFFSET	SIZE	WRITE	STATUS			
0	0	34	20000	LATE	AVAIL			
DRIVE	CYLS	HEADS	SECTORS	BLOCKS	MTYPE	OPT	FAIR	STATUS
0	1224	5	34	291312	FIXED	NONE	24	SPUN UP

Drive number :

Enter the drive number (zero on a single-drive system). WOMBAT will then display the Master Menu options.

```
** Master Menu **  
1  Structure Disc  
2  Test Disc  
3  Manage bad blocks  
4  Display error  
5  Shadow options
```

Select an option by typing the option number followed by RETURN. Options 1 through 3 and 5 will provide sub menus while Option 4 displays the last controller detected fatal error. To return to the master menu from a sub menu type RETURN.

To exit from the master menu to the announcement (to select a different drive) type RETURN. WOMBAT will not allow you to do this before verifying whether the disc structure data has been written to disc. 'NO' is the default value.

### 3.3 Master Menu Options

#### Option 1 - Structure Disc

Selecting this option causes a sub menu to be displayed as follows:

```
** Disc Structure Menu **  
  
1  Create Disc Structure  
2  Format Disc  
3  Write Disc Structure  
4  Update Header Blocks  
5  Display Disc Structure  
6  Change Unit number  
7  Display Statistics
```

#### Option 2 - Test Disc

Selecting this option causes a sub menu to be displayed as follows:

```
** Disc Test Menu **  
  
( ! means all data on Disc destroyed )  
  
1  Read All Disc (preserves all data)  
2  ! Write Disc !  
3  ! Pattern Test !  
4  ! Random Writes !  
5  ECC Validation  
6  Read Physical Block  
7  Display Error Statistics  
8  Zero Error Statistics  
9  Test Cache, RAM & ROM
```

### Option 3 - Manage Bad Blocks

Selecting this option causes a sub menu to be displayed as follows:

**\*\* Bad Block Management Menu \*\***

- 1 Manually Replace Bad Block
- 2 Automatically Replace Bad Blocks
- 3 Display Replaced Bad Blocks
- 4 Enter defect map
- 5 Get defect map from drive

### Option 4 - Display Error

Selecting this option causes WOMBAT to display a message which explains the most recent controller detected fatal error. Refer to Table 4-6. The occurrence of an error is indicated by the controller hanging with its red LED flashing. If no error has occurred this option produces a meaningless error message.

### Option 5 - Shadow Options

Selecting this option causes a sub menu to be displayed as follows:

**Shadow options menu**

- 1 Shadow copy option
- 2 Set shadow units
- 3 Reset shadow units
- 4 Copy unit to unit
- 5 Compare unit to unit

### Setting Up a New Disc

The procedure for structuring a new disc is as follows:

1. Create disc structure.
2. Format the disc.
3. Write the disc structure.
4. Pattern test the disc.
5. Replace bad blocks.

Once these 5 steps have been undertaken the host operating system may use the disc.

### 3.4 Disc Structure Menu Options

#### **\*\* Disc Structure Menu \*\***

- 1 Create Disc Structure
- 2 Format Disc
- 3 Write Disc Structure
- 4 Update Header Blocks
- 5 Display Disc Structure
- 6 Change Unit number
- 7 Display Statistics

#### **Option 1 - Create Disc Structure**

The Create Disc Structure option must be performed when a new disc is connected to the WQESD. This allows the various disc geometry, controller wide tuning parameters and virtual unit structure to be specified. The virtual unit structure allows a single large drive appear to the host operating system as multiple drives.

This option enters an interactive question and answer dialogue which specifies the disc structure. WOMBAT displays either the current or the default value for a parameter and gives you the option of accepting or changing it to a new value. To accept the displayed value, hit RETURN. To change it, type in the new value followed by RETURN. If WOMBAT detects improper values it will issue a warning.

The create disc structure dialogue is divided into three parts.

- (a) The drive structure specification, which describes the physical geometry of the drive.
- (b) The unit structure specification, which is executed once for each virtual unit defined where the size of each unit and unit specific parameters is described.
- (c) The controller wide tuning parameters, where read lookahead and command queue size are specified.

#### **Drive structure specification**

**Cylinders:** The number of cylinders on the drive. There is normally no need to change this value. Note that this will be one less than the number of cylinders specified by the manufacturer - this is to preserve the manufacturer recorded defect list on the last cylinder. You may override this option and recover the last cylinder, but the manufacturers defect list will be destroyed during formatting.

- Heads:** The number of heads on the drive is displayed. There is normally no need to change this value.
- Full sectors/track:** Hard sector drives report the correct value here. For soft sector drives, the required value will have to be computed and entered. See section 3.12 - Computing Sectors Per Track.
- Track spiralling:** Track spiralling improves disc performance on data transfers over more than one track. Sector zero on each track is offset by a nominated factor to allow head select and positioning before sector zero on the next track is reached. The recommended factor is four. Note that the Track Spiralling factor must not be changed after the disc is formatted. If it is changed, incorrect bad block replacement will occur. If you wish to try a different Track Spiralling factor to that with which the disc is currently formatted, then the disc **MUST** be reformatted with the new Track Spiralling factor.
- Optimisation:** The seek optimisation strategy can be either:
- 0 (None) - No optimisation done. First request found executed. This may not be the next sequential request.
  - 1 (Nearest) - Selects request that is closest to the to the current cylinder.
  - 2 (Elevator) - Processes requests as it moves in one direction along the disc until it reaches the last request in that direction. This means that "Elevator" favours the centre of the disc, as it passes it twice as often as the periphery.
  - 3 (Forward) - This processes requests from the lowest cylinder number to the highest in one direction only.
- Note that optimisation is only effective if the host operating system supports multiple accesses. RT-11 and TSX plus do not support optimisation without a special device handler.
- Fairness count:** The fairness count determines the number of times an I/O request will be passed over by the controller's seek optimisation setting before it is executed. A reasonable count for normal use would be around 25. Every time a request is passed over its fairness count is decremented. When that count reaches zero that request will

be selected, no matter what optimisation strategy is in effect. This count has no effect if no optimisation is selected.

**Removable Media:** The Removable media option should be set if removable media drive(s) are installed. If removable media is specified, then the drive can not be partitioned. Also, the drive number and unit number are always the same. This is necessary because WOMBAT has to be able to determine the unit number if the media is removed.

If removable media is specified, WOMBAT will not ask for a unit number.

This completes the drive structure specification. WOMBAT next prompts for a unit number (if removable media is not being used). Type the unit number of the next unit to be defined, then hit RETURN. When the unit is completely defined, WOMBAT will again prompt for a unit number. If there are no more units to define hit RETURN. WOMBAT will then proceed to the controller wide parameter definition. Note that if no units are defined, the operating system will not see anything attached to the controller.

### Unit Structure Specification

**Unit size:** If an existing unit number is specified WOMBAT will display its size in blocks. If a new unit number is specified WOMBAT will display the size in blocks of the first unallocated disc area it finds beginning at the start of the disc. On a new disc this will be the entire user area. This can be changed to a smaller value if necessary. To delete an existing unit, specify zero for this field.

**Media type:** This field is displayed by some operating systems when you enquire about the type of drive. As a part of unit status when a "Get Unit Status" command is issued the MSCP protocol returns a 5 character media type. The first two characters must be 'DU'- for example - DURD54. To change this enter 1 to 5 alphabetic characters and 2 digits, e.g. RD52, to emulate DEC's 31 megabyte Winchester. For example, RSX-11M-PLUS responds to a "DEV DU:" command with : "DU0: Public Mounted Loaded Label = RSX11MPBL15 Type = RA81"



**Serial number:** The MSCP protocol returns a 32-bit volume serial number as a part of its response when an "on-line" command is issued. WOMBAT defaults this field to zero. To change this enter the desired serial number. This field is used, for example, by RSX-11M-PLUS, when a disc is initialized with the "INI DU:" command. It sets up the volume serial number.

**Host write confirmation:** This specifies if the controller is to notify the host that a write request has been completed. Enter 1 if the host is to be notified when the data is in the cache. The data will be written to the disc later. Enter 2 if the host is to be notified when the data has actually been written to the disc.

This completes the unit structure definition. Now WOMBAT enters the final part of the dialogue which specifies the controller wide tuning parameters.

#### Controller Wide Tuning Parameters

**Read lookahead:** This is a feature of the cache which allows the controller to read a specified number of sectors in addition to those requested by the host. Enter the minimum number of blocks you wish the controller to read for any request. For example if a value of 4 is specified, when the host asks for a single block to be read, the next 3 blocks will automatically be read into the cache. If the host subsequently asks for one of these 3 blocks then the request can be honoured immediately from cache. If the host requests a transfer equal to or larger than the read lookahead size then this parameter will have no effect.

**Cache Enable:** This parameter enables or disables the cache. A value of 0 disables the cache and a value of 1 enables the cache. The cache cannot be selectively enabled or disabled for a particular drive or unit number.

**Command queue:** This parameter allows you to specify the number of commands the controller can stack. The controller will then attempt to optimise the order in which they are executed. Large command

queue stacks incur considerable overhead and will degrade controller performance. Note also that some operating systems (RSX-11M-Plus 2.1B is a good example) have a maximum limit for the size of the stack. The default size [8.] is a good compromise and is acceptable to most operating systems.

**Front Panel Type:** This option allows the correct selection of front panel type. See 2.4 Front Panel Connections.

- 0 - None
- 1 - Passive
- 2 - Active

This completes the disc structure definition. WOMBAT now checks the tables for consistency and returns to the disc structure menu.

#### Option 2 - Format Disc

WOMBAT asks you to confirm this drastic action as it will destroy ALL data that resides on the disc. WOMBAT will then initiate a two pass formatting operation. During the first pass WOMBAT creates all the sectors on the disc, write a sector headers which contain the sector number, the head number, and the cylinder number, as well as preambles and sync bytes, followed by a 2 byte data field. During the second pass WOMBAT writes a test pattern to each sector, preparing the disc for read testing. WOMBAT then writes the disc structure onto the reserved areas.

#### Option 3 - Write Disc Structure

WOMBAT will ask you to confirm this drastic action as any existing disc structure will be destroyed. WOMBAT will then write the new structure onto special reserved areas of track zero. The data is recorded twice for improved recoverability. A total of 6 blocks is written on track zero. After the structure has been written, the drive's replacement block table is zeroed. If there were any replaced blocks recorded there, they will be lost. However they will still be marked as replaced and will generate hard errors during a read operation.

#### Option 4 - Update Header Blocks

This is similar to Write disc structure except that the replacement block table is not written, thus preserving any blocks which may have been replaced.

## Option 5 - Display Disc Structure

WOMBAT displays the structure of the currently selected drive in a form similar to the create disc structure dialogue. This is useful for checking that the newly created structure is correct.

## Option 6 - Change Unit Number

It is sometimes necessary to change a unit number in order to resolve a duplicate unit number or to satisfy operating system requirements. This method is a safe and simple way of doing so. WOMBAT prompts for a unit number on the current drive, and then for the new number for that unit.

## Display Statistics

Statistics about disc and cache usage are maintained, and recorded on the disc once per 1000 disc accesses. They are displayed as :

### Controller statistics report

# of commands	xxxx	
# of reads	xxxx	# of cache hits xxxx (xxx%)
# of writes	xxxx	

### Drive statistics report

Drive #	Soft errors	Re-vectors	Blocks replaced
xxx	xxxx	xxxx	xxxx
Drive #	Seek distance	# of seeks	Seek errors
xxx	xxxx	xxxx	xxxx

**Commands** is the number of MSCP commands issued.

**Re-vectors** is the number of accesses to replaced blocks.

**Seek distance** is the total seek distance, in cylinders.

**Blocks replaced** is the number of blocks dynamically replaced by the controller during normal operation, rather than through WOMBAT.

**Reset Counters** is then asked. "Y" will reset them to zero.

### 3.5 Disc Test Menu Options

A disc can be tested after it has been formatted and before the structure is written to it. Testing does not overwrite the HDR or RCT blocks. The disc structure must be written to the disc before bad blocks can be replaced.

All tests continue indefinitely until aborted by one of the following methods:

1. If an RS232 serial port terminal is attached to the controller, press BREAK.
2. If WOMBAT is running from the Console terminal, type CTRL/C.

When a test is aborted the Test Disc Menu options are returned. If tests are run from an RS232 terminal attached to the controller, beware of system activity on the host computer as Qbus initializations will cause the disc controller firmware to re-initialize and so leave WOMBAT.

All tests give 10 retries on an error, reporting every error by displaying the block number and an error code.

#### **\*\* Disc Test Menu \*\***

( ! means all data on Disc destroyed )

- 1 Read All Disc (preserves all data)
- 2 ! Write Disc !
- 3 ! Pattern Test !
- 4 ! Random Writes !
- 5 ECC Validation
- 6 Read Physical Block
- 7 Display Error Statistics
- 8 Zero Error Statistics
- 9 Test Cache, RAM & ROM

#### **Option 1 - Read All Disc**

This test reports any read errors. Successful operation will be reported in the following format:

Pass: 1. Errors: 0.  
Pass: 2. Errors: 0.

This function does not destroy any information.

### Option 2 - Write Disc

This test reports any write errors while writing a test pattern to the whole disc. ALL INFORMATION on the disc, excepting HDR and RCT blocks, is DESTROYED. Errors are displayed in the standard format:

```
Block: 32040      (Error message)
Pass:1. Errors: 1.
Pass:2. Errors: 1.
```

The displayed error count is cumulative until the test is terminated.

### Option 3 - Pattern Test

A test pattern is written to each block. WOMBAT does one write and 10 read passes. This test reports any errors in the standard format as shown above.

### Option 4 - Random Writes

This test writes 5000 blocks at random locations in the user area of the disc. It then reads the entire disc to determine if any of the writes caused an error. This test is designed to test the head positioning and selecting logic of the drive.

### Option 5 - ECC Validation

The ECC test uses a special reserved block on track zero for testing. It first proves that it can successfully correct an 11 bit error and then proves that it cannot correct a 12 bit error. This test checks the ECC logic within the WQESD.

### Option 6 - Read Physical Block

WOMBAT prompts for a block number anywhere on the disc. It then converts that block number into a physical address consisting of cylinder, head, and sector, and displays these values. Then it reads that sector and displays a message indicating the success or failure of the read.

### Option 7 - Display Error Statistics

Displays the error statistics gathered by any of the above disc testing options in the following format:

```
** Error Statistics **
Block   Number (of errors)
32040   1.
Blocks in error: 1.
```

### **Option 8 - Zero Error Statistics**

Zerues the error statistics table & redisplayes Test Menu options.

### **Option 9 - Test Cache, RAM & ROM Option**

This test continuously writes test patterns throughout the entire cache and reads them back testing for veracity. A separate part of the test automatically checks that the parity logic is functioning correctly by forcing incorrect parity and checking that an error occurred. The cache pattern tests use special microcode instructions which iteratively read and write large blocks of cache memory at high speed. The Static RAM at L5 is also tested and the code PROM is Checksummed.

## **3.6 Bad Block Management Menu Options**

### **\*\* Bad Block Management Menu \*\***

- 1 Manually Replace Bad Block**
- 2 Automatically Replace Bad Blocks**
- 3 Display Replaced Bad Blocks**
- 4 Enter defect map**
- 5 Get defect map from drive**

### **Option 1 - Manually Replace Bad Blocks**

WOMBAT prompts for a block number within the user area of the disc. Then it marks the specified block as bad and allocates a replacement block for it.

### **Option 2 - Automatically Replace Bad Blocks**

WOMBAT searches the error statistics table, which is compiled by the read, write, and pattern tests, for blocks whose error count exceeds three. Any such blocks are marked as bad on the disc and replacement blocks are allocated for them.

### **Option 3 - Display Replaced Bad Blocks**

WOMBAT reads the Replacement Control Table and displays the logical block numbers of any blocks recorded there.

#### Option 4 - Enter Defect Map

This option prompts for the drive manufacturer's defect map information as follows :

Enter defect map for drive n

Enter all values in decimal

Bytes per sector:           The number of bytes per sector as set up in the drive switches (hard sector drives), or calculated by following the procedure in section 3.12 Computing Sectors per Track.

Cylinder:                   Cylinder number of defect, RETURN or ^C if no more.

Head:                       Head number of defect.

Bytes past index:           Location past index of defect.

Bit length of defect:       Length of defect in BITS.

This data is then used to compute the address of a block on the disc. If it does not match a block on the specified disc track the error message "!! Beyond last sector" is produced. This may mean that the defect is located beyond the last data sector on the track, or that the entered data was wrong. WOMBAT then calculates the block number, displays it and replaces it. The "Cylinder:" prompt is then repeated.

#### Option 5 - Get Defect Map from Drive

This option tells the controller to find then read the defect map information written on the drive by the manufacturer. It will then list the defective blocks by head number and ask for confirmation of replacement. It is important that the displayed data be checked, as it may have been corrupted by previous formatting/overwriting. . WOMBAT does do some checks, but relies partially on the operator checking the validity of the list, in particular, the fields underlined below.

ESDI defect map for drive n

Soft/Hard sectored drive, xxx bytes/sector

Defect map read error, head x	Displayed	if	the
	defect	map	is
	unusable.		

Then, for each head on the drive :

Defect map read: Date - mm-dd-yy Fujitsu/Maxtor model #

Head x

Defect list for head x

Cyl. xxx, bytes past index xxx, length xxx      Displayed      for      each  
error.

When all the defects have been displayed,

Do you wish to proceed with replacement for head x?

Only reply "Y" if the list appears correct. A defect at cylinder 20,000, 30,000 bytes past the index, and of length 0 is a possible (but patently wrong) display.

The defects are again displayed as they are replaced :

Cyl. xxx, bytes past ind. xxx, length xxx: block xxx, byte xxx

When this is complete, the message

Defect map replacement complete for head x

is displayed, and the data for the next head processed.



### 3.7 Shadow Options Menu Option

This feature shadows, i.e. keeps two copies, of a 'logical unit'. When the disc controller and drives are first powered up, any unit (and a physical disc may be broken up into a number of 'logical units') that is shadowed has its contents copied to its shadow unit. Thereafter any update of that unit will cause the controller to update the shadow unit as well. When reading from a shadowed unit, the drive with its heads nearest the required data is used. This helps to keep the drive shadowing overhead down, although in normal circumstances, with writes consisting about 10% of reads, there is a performance penalty.

Shadowing consists of two operations: the initial copy of the entire primary unit to the shadow unit; and then the updating of both primary and shadow units on every disc write. Updating of both copies takes place as a matter of course, while the initial copying can happen at various times depending on the 'Shadow copy option' selected.

The main reason for drive shadowing is RELIABILITY. When reading, any error detected will cause the controller to use the data on the other drive. This means that, besides the controller having extensive error recovery facilities, there is a redundant backup of all data on the shadowed unit without any user programming or operating system overhead.

Drive shadowing is completely controlled from the 'Shadow options' menu.

#### Shadow options menu

- 1 Shadow copy option
- 2 Set shadow units
- 3 Reset shadow units
- 4 Copy unit to unit
- 5 Compare unit to unit

### Option 1 - Shadow Copy Options

The first step is to select how the unit (the 'Primary unit') is to be copied to its 'Shadow unit'. There are 4 options :

- |                                 |   |
|---------------------------------|---|
| 0 - WOMBAT Only                 | This is the default. With this selected the primary is only copied to its shadow when you invoke it in the Copy Unit to Unit option.                                |
| 1 - On power up                 | This causes the copying to be done whenever the drive(s) are first powered up, or WOMBAT has been invoked.  |
| 2 - Not ready/ready             | The copying is to be done whenever the controller detects the shadow unit going from a not ready to ready condition - e.g. a removable drive being brought on-line. |
| 3 - Power up or Not ready/ready | A combination of 1 and 2 above.   |

The use of copy type 0 only is recommended at this time. It is considered that the other options have too high a probability of copying bad data from the primary unit over good backup data on the shadow unit.

### Option 2 - Set Shadow Units

Shadow units are logically connected to primary units by this option which asks :

- |               |                                |
|---------------|--------------------------------|
| Primary unit: | The unit to be shadowed.       |
| Shadow unit:  | The unit number of the shadow. |

The shadow unit will not be available to any operating system as it appears as 'unit undefined'. It must be EXACTLY the same size as the primary unit.

Now copy the primary to its shadow by using the **Copy unit to unit** option.

You may shadow a unit on the same physical drive, though you lose both reliability, by having the data on only one drive, and performance, by having the disc heads move to write all data twice.

If you have selected a shadow copy type of 1 or 3, whenever the controller is powered up and its first 'MSCP Initialize' sequence executed for the primary unit (the unit you wanted shadowed), the contents of this unit are copied to its shadow, or if WOMBAT has been invoked. The time this takes obviously depends upon the number of blocks on the unit, but we have found that a third of a megabyte

per second is a good guide. While this is happening user I/O may take place, but will be made slower by the 'shadow copy' taking place.

**WARNING:-** a potentially serious problem exists with the automatic copy of data from the primary to the shadow at power up (shadow copy types 1 or 3). If the primary has been corrupted, or the data is invalid, IT WILL DESTROY YOUR BACKUP. If you have trouble with your primary unit use WOMBAT to change it from a shadow primary. You may even want to change its unit number via the 'Change Unit number' option, and then bring in the shadow unit as its replacement. Please realize that the controller MUST assume the validity of the primary unit at the first initialization after power up if you have selected one of the power up copy types.

### Option 3 - Reset shadow units

This option allows disabling of an assigned shadow unit :

**Primary unit:**           The primary unit.  
**Shadow unit:**           The unit number of the shadow to be  
                              disconnected from the primary.

This shadow unit will now be available to the operating system.

### Option 4 - Copy Unit To Unit

To copy a primary unit to its shadow :

**Primary unit**            The number of the unit you want copied.  
**Shadow unit**            The unit you want the data copied to.

The copy will take place. Data is transferred at about twenty megabytes a minute

### Option 5 - Compare Unit to Unit

This option allows the comparison of the data on a primary and a shadow unit.

**Primary unit**            One of the units.  
**Shadow unit**            The other.

If any data errors, or data compare mismatch, is found, the block number and type of error will be reported. The comparison proceeds at about two megabytes a minute.

### 3.8 WOMBAT Disc Structure

WOMBAT records the logical structure of the drive on track 0. All of track 0 is reserved for this and other testing purposes. The user area begins at the next block after track 0. This is the same as the number of sectors per track. The user area extends to the block before the beginning of the Replacement Control Table (RCT). The size of the RCT is fixed and is always 2 tracks. The 2 tracks are accessed by different heads and contain identical data. All the blocks from the end of the second RCT track to the end of the media are reserved for replacement blocks.

The disc area reserved for RCT and replacement blocks is always an integral number of cylinders. The number of cylinders required is always computed from the total number of cylinders on the drive. This reserves a constant proportion of the media for replacement blocks. The amount reserved is approximately 0.1% of the total formatted capacity. Option 4 of the master menu will report the formatted capacity of each drive. You can determine the user area easily. It will be reported as the size of the first unit configured on that drive. If there is more than 1 unit, then the sum of their sizes is the total user area, assuming there are no unallocated areas on the disc.

All block numbers in WOMBAT are physical block numbers beginning at the first sector of the first head of the first cylinder of the drive, which is defined as block zero. The last block on the drive is block  $n-1$  where  $n$  is the total number of blocks on the drive. Therefore, the first block of the user area is not block zero. Its block number is the same as the number of sectors per track, as track 0 on the drive is reserved. Option 4 of the master menu will display both the size and the offset (starting block number) of each unit defined. Using these figures you can determine the exact position and extent of any unit.

### 3.9 Drives with Removable Media

Some manufacturers offer drives with removable media. The WQESD detects when such a drive is connected and makes certain changes to the way it operates. The important changes are that you can only have one virtual unit on a drive whose media can be removed. That unit must have the same unit number as the physical drive number. The reason for this is that it is necessary to determine the unit number even though the media may be removed. This means that the unit number cannot be recorded on the media for these drives. WOMBAT automatically detects these drives and prevents you from creating incorrect units on them.

### 3.10 Error Recovery Procedures

When a data error occurs the transfer is retried 10 times. If any of the retries is successful, then the block is not replaced. Otherwise, the drive is recalibrated and a further 10 retries are executed. If any of these succeeds then the block is not replaced. If not, the block is assumed to have a hard error. When a hard error is detected the error correction algorithm is invoked and dynamic replacement is scheduled.

When a hard error is detected, a quick check is made to verify that the problems were not caused by a drive hang. A seek to cylinder zero is performed followed by a read of block zero. Then a seek to the RCT cylinder is performed followed by a read of the first block of the RCT. If all of this succeeds then the drive is deemed to be 'OK' and the controller then proceeds with dynamic replacement.

When it has decided to replace a block, the controller first writes the data from the bad block to a special area of the rct together with a flag to indicate whether or not ECC was successful in recovering the data. This transfer is done as a write check operation. Next the bad block is tested by write checking data patterns to it a number of times. If the block appears good after this test, then it is assumed that the error was transient and the original data is written back to the block. The block is not replaced. If however the block is confirmed as being bad, then a new block is allocated to replace it, and the previously saved data is written to it.

When the saved data is written to the replacement block, the ECC success flag is tested. If it indicates that the data was unrecoverable then the block will be written with forced error status.

### 3.11 Drive shadowing

The intent of this feature is to provide a continuous automatic backup of important data. You must first specify a unit to be the shadow unit, that is, the unit onto which the data will be backed up. The primary unit can then be specified - this is unit which the host will see. The normal case would be to have two identical drives with a single unit on each. The primary unit must be of identical size to the shadow. Once the primary/shadow pair has been defined, then the shadow unit will become invisible to the host. All writes directed to the primary will also be written to the shadow. If a read is directed to the primary, the controller will attempt to redirect that read to whichever drive of the pair has its heads positioned closest to the requested data.

This feature will only offer protection against the primary unit failing. It cannot protect, as a normal backup can, against a rogue program which writes garbage on the disc. The garbage will simply be copied onto the shadow unit as well duplicating the garbage. It is advisable, although not mandatory, that the shadow unit is on a different drive to the primary if the maximum protection is to be gained.

If the primary does fail, then you will want to recover the data from the shadow. To do this WOMBAT must be run. First, disable the shadow unit with the "Reset shadow units" option of the Shadow Options Menu. Then (optionally) change the unit numbers of the two units with the "Change unit number" option of the Disc Structure Menu. Once the data has been recovered, the primary unit can be serviced and re-installed in the system.

### 3.12 Computing Sectors Per Track

To compute the number of sectors per track first determine the number of bytes per track from the drive manual. Divide this figure by the number of bytes per sector. You determine the number of bytes per sector by the following calculation:

$$\text{Bytes per sector} = K + (\text{PLO} * 2)$$

where: K = 566 for hard sectored drives  
K = 557 for soft sectored drives

PLO is the number of bytes of PLO sync required by the drive. Obtain this figure from the drive manual.

e.g. Maxtor EXT4125, PLO sync field is 26 bytes

$$\begin{aligned}\text{bytes/sector} &= 557 + (26 * 2) \\ &= 609 \text{ bytes per sector}\end{aligned}$$

This drive has 20808 bytes/track minimum

$$\begin{aligned}\text{sectors/track} &= 20808 / 609 \\ &= 34.17\end{aligned}$$

so choose 34 sectors per track

e.g. Hitachi DK512 PLO sync field is 11 bytes

$$\begin{aligned}\text{bytes/sector} &= 566 + (11 * 2) \\ &= 588 \text{ bytes per sector}\end{aligned}$$

This drive has 20832 bytes/track minimum

$$\begin{aligned}\text{sectors/track} &= 20832 / 588 \\ &= 35.42\end{aligned}$$

so choose 35 sectors per track

### 3.13 WOMBAT Error Messages

The following is a list of the error messages displayed by WOMBAT.

sector not found	Indicates that the sector asked does not exist or cannot be located.
drive fault	Indicates that the drive is faulty and that service is necessary.
drive timed out	Indicates that the drive has failed to complete an operation.
data field error	Indicates that bad data exists in a sector.
controller fault	Indicates controller failure. Service will be necessary.
block marked as bad	Indicates that the block has been flagged as bad. The controller will refer to the RCT for a replacement

	block.
data late	Data is lost due to internal overflow in controller memory before transmission over the bus to host.
forced error	A forced error occurs when a good block has bad data. The block is flagged forced error until good data is written to the block.
seek error	An error has occurred on a seek operation.
rct full	The replacement control table is full. An error of this kind indicates that the disc has too many errors to be serviceable.
rct read error	Fatal error which indicates the controller cannot read the RCT.
rct write error	Fatal error which indicates the controller cannot write the location of replacement blocks.
illegal sector specified	A sector with an illegal number has been specified.
illegal block number	A block with an illegal number has been specified.
non existent drive	A drive has been specified which does not exist.
unit table full	The unit table is full. The maximum number of units is 16. To create a new unit an existing unit must be undefined.
no drive selected	A drive must be selected before WOMBAT can perform any operation. Select any valid drive.
non-existent unit	The unit selected has not been defined.
disc structure write error	WOMBAT is unable to write the structure to the disc. This is a fatal error indicating that the drive is not serviceable.



### 3.14 WOMBAT Self-Diagnostics

#### Initialization procedures

A common initialization procedure exists for both WOMBAT and the MSCP firmware. It performs :

- a RAM integrity test
- a ROM checksum
- various checks on the disc drive and its structure

The errors which can result from this are described under Section 4.9 Fatal Controller Errors.

## Chapter 4 WQESD MSCP Programming

### 4.1 Overview of MSCP

Mass storage control protocol (MSCP) is a message-oriented set of rules by which the WQESD controller module and the host system communicate. This protocol allows the host to send message requests for data reads or writes to the controller and receive response messages back from the controller. The host does not concern itself with details such as device type, media geometry, media format, or error recovery.

All software and hardware functions are partitioned into independent 'host' and 'controller' layers. Each layer consists of a high-level I/O driver and a communications server. The controller layer receives and processes commands which have been formed by the host layer.

The communications server handles all communications protocol between the I/O layers, leaving the I/O system free to process data requests. Communications between host and controller are carried out on the I/O bus without having to generate processor interrupts. The host's communications server monitors all command transmission and response and in the event of failure or error, initiates recovery procedures.

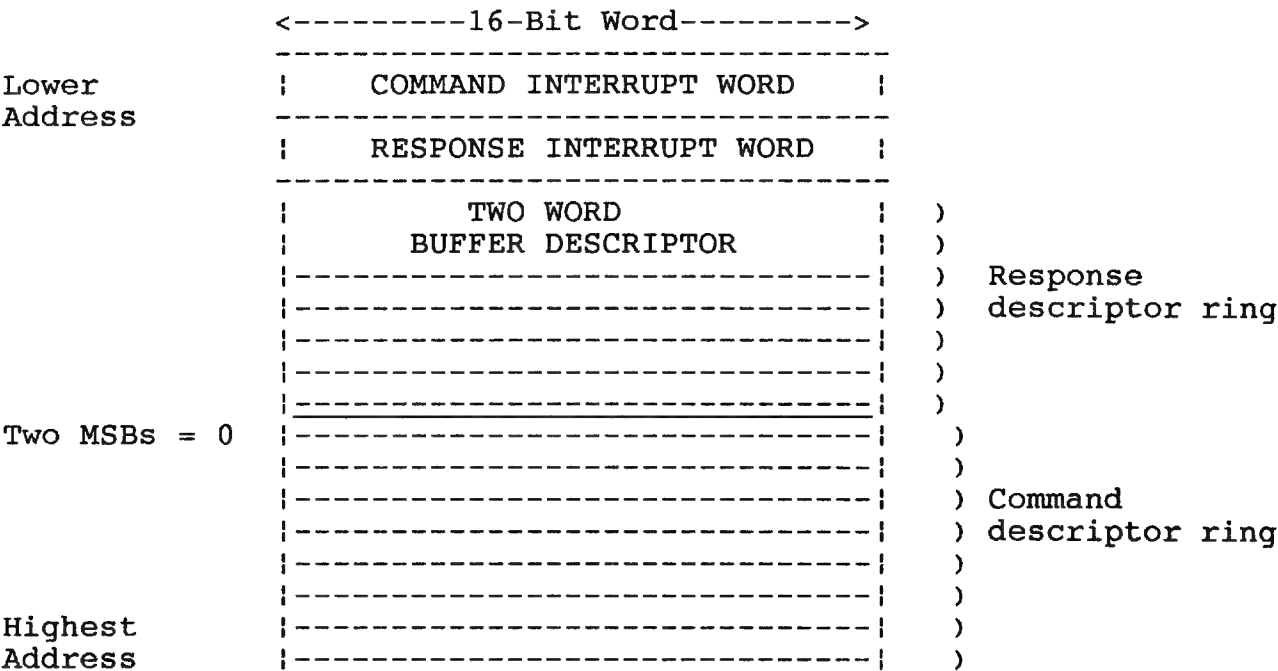
Disc drive parameters are transparent to both the host and controller resident layers of MSCP. The disc drive passes factors such as disc geometry, storage capacity or error retry counts to the disc controller on system start-up.

In addition to relieving the host of disc-specific data, the disc controller and disc provide the host with "clean" data. The disc drive handles some positioner errors entirely by itself but performs certain error-recovery operations under direction of the disc controller.

### 4.2 Controller Communications

The host designates an area of memory to be used as a communications area between itself and the controller. This area is made up of two sections a header area containing interrupt identification words and a variable-length section containing the response (receive) and command (send) lists, organised into ring buffers. Figure 4-1 shows the memory communications format.

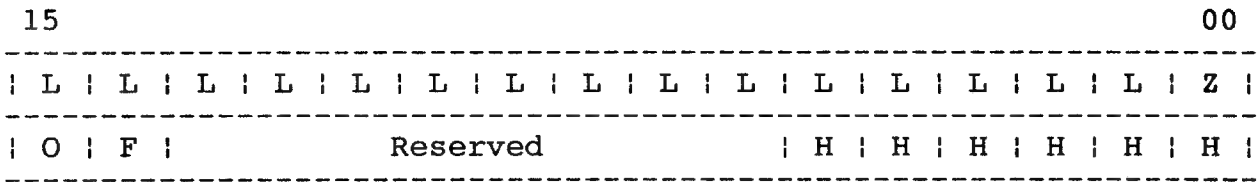
FIGURE 4-1    MEMORY COMMUNICATIONS FORMAT



Command and Response Rings

Command and response lists are organised into 'rings' of 32 bit descriptors. The length of each ring is determined by the speed with which the host and controller generate and process messages. The host sets the ring lengths at initialization time. Figure 4-2 describes the Descriptor Format while Table 4-1 details the code descriptions.

FIGURE 4-2    DESCRIPTOR FORMAT



**TABLE 4-1    COMMAND RING CODE DESCRIPTIONS**

---

CODE	DESCRIPTION
<hr/>	
Z	Is zero, as envelope address (text+0) is word-aligned. The controller will always assume that Bit 00 is set to zero.
L	Low-order envelope address.
H	High-order envelope address.
F	Flag bit.

---

When the controller returns ownership to the host it sets F=1 to indicate that it has completed action on the descriptor.

When the controller acquires ownership of a descriptor from the host, F=1 indicates that the host is requesting a ring transition interrupt. If F=0, the host is not requesting a ring transition interrupt. An interrupt will occur only if this descriptor causes a ring transition and if transition interrupts were enabled during initialization.

The controller always sets F=1 when returning a descriptor to the host, so if a host wishes to override ring transition interrupts it must always clear F when passing ownership of a descriptor to the controller.

0 Ownership bit. Set to 0 if owned by the host or 1 if owned by the controller. Interlocks the descriptor against premature access by either party.

### Message Packets

The command or response descriptor points to word (text+0) of a 16-bit word-aligned message envelope formatted as shown in Figure 4-3. Table 4-2 describes the word envelope contents.

FIGURE 4-3 MESSAGE ENVELOPE FORMAT

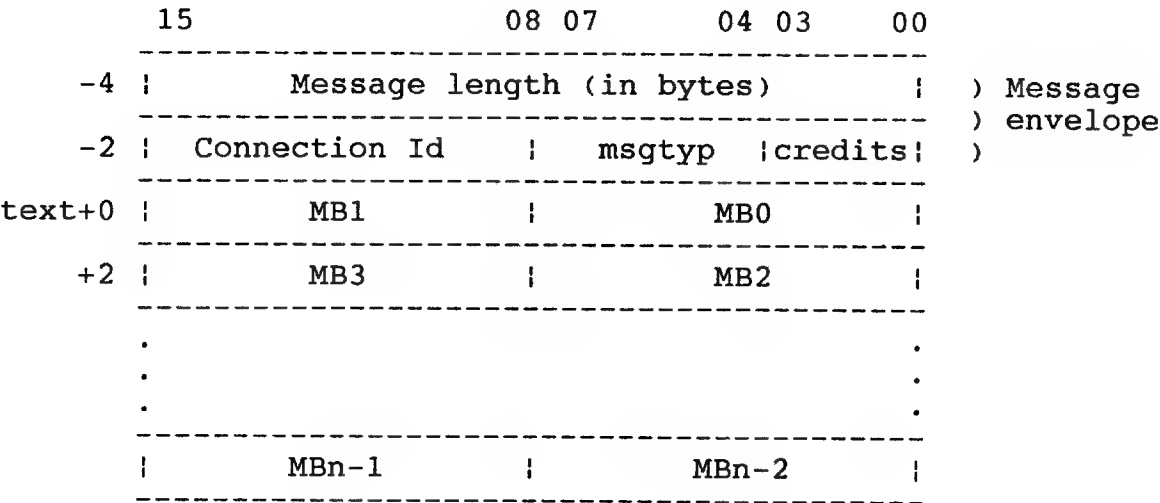


TABLE 4-2 WORD ENVELOPE CONTENTS

---

WORD	ENVELOPE CONTENTS
------	-------------------

---

0	<p>Message length, in bytes.</p> <p>For commands, this length is equal to the size of the command (in bytes), beginning with [text+0].</p> <p>For responses, the host sets the length equal to the size of the response buffer (in bytes), beginning with &lt;text+0&gt;. Before actual transmission of a response, the controller reads the field length in the message envelope. If the controller's response is longer than the response buffer, the controller will fragment its response into as many response buffers as necessary.</p> <p>The controller sets the resulting value into the message length field. The host must therefore keep re-initializing the value of this field for each proposed response. If a controller's responses are less than or equal to 60 bytes, then the controller need not check the size of the response slot.</p>
1	<p>Connection Id</p> <p>Identifies the connection serving as a source of, or destination for, the message in question.</p>
2	<p>Message Type</p> <p>The following response ring message types are implemented:</p> <ul style="list-style-type: none"><li>MSGMNT Maintenance packet (diagnostic)</li><li>MSGCRD Credit notice (ignored)</li><li>MSGDAT Datagram packet.</li><li>MSGSEQ Sequential packet</li></ul>
3	<p>Credit field</p> <p>Gives a credit value (usually one) associated with the message. This mask, in response packets, is added to the controller's credit field to give the number of commands-in-progress. So while Word 1 is always 1 for the command ring, this is not the case for response rings.</p>

---

### 4.3 Message Transmission

#### Command Transmission

When the ownership bit (O) of a command ring descriptor is equal to 1, it means that the host has filled the descriptor and is releasing it to the controller. When the ownership bit (O) resets to zero, it means that the controller has emptied the command ring descriptor and is returning ownership of the descriptor to the host.

To ensure that the controller sees every command, the host must read the IP register whenever it inserts a command in the command ring. This forces the controller to poll the command if it was not already accessing the command ring.

#### Response Transmission

When the ownership bit (O) of a response ring descriptor is equal to zero, it means that the controller has filled the descriptor and is releasing it to the host. When the ownership bit (O) sets to 1 it means that the host has emptied the response ring descriptor and is returning ownership of the descriptor to the controller. Just as the controller must poll for commands, so must the host poll for responses.

#### Interrupts

The transmission of a message will result in a host interrupt from the controller under the following circumstances.

1. During the initialization process (open a 'connection').
2. When the command ring buffer transitions from 'full' to 'not full'. This interrupt means that the host may place another command in the command ring.
3. When the response ring buffer transitions from 'empty' to 'not empty'. This interrupt means that there is a response for the host to process.

4. When a fatal controller error is detected and an interrupt can be generated. These are:

- Failure to become Qbus master for data transfer
- Failure to become Qbus master for interrupt
- Failure to access I/O page registers or communication area
- Qbus parity error detected.

#### 4.4 Data Transmission

In the command ring, the descriptor points to a command packet. Within the command packet is a buffer descriptor which contains a pointer and a byte or word count. The buffer descriptor points to the data buffer which holds data transfers. The data is moved by the controller into or out of the buffer as DMA transfers to/from Qbus addresses.

#### 4.5 Initialization

The purpose of initialization is to identify the parameters of the host-resident communications region to the controller, provide a confidence check of controller integrity, and bring the controller online to the host.

##### Initialization Process

This paragraph describes the activity within the SA register during an initialization process. This is dependent on whether SA is being read or written.

By moving 4000 into IP, the controller initializes and passes back the 'step' response in SA. Then, the initialization parameters are written into SA. There are 4 words of initialization, and the controller must reflect each step by the appropriate step response, which is also returned in SA. The initialisation parameters are set out below in Table 4-3.



**TABLE 4-3     INITIALISATION PARAMETERS**

---

WORD	CONTENTS
<hr/>	
0	Command and Response ring sizes, interrupt enable and vector. The host writes into SA the lengths of the rings, whether interrupts are to be armed, and if so, the address of the interrupt vector. The controller then runs a complete internal integrity check and signals either success or failure.
1	Low order address of communications area, ie., ring buffer address. The host reads an echo of the ring lengths from SA, and then writes into SA the low-order portion of the ring base address.
2	High order address of communications area, bits 0-14. The interrupt vector address and the master interrupt arming signal are echoed in SA. The host then writes the high order portion of the ring base address to SA along with a signal that conditionally triggers an immediate test of the polling functions of the controller.
3	Burst transfer control, last failure flag, and the 'GO' bit. The controller tests the ability of the Qbus to perform DMA transfers. If successful, the controller zeros the entire communications area, and then signals the host that initialization is complete.

---

#### 4.6     Registers

The programmable registers contained on the WQESD are the Initialize and Poll register (IP) and the Status and Address register (SA).

##### Initialize and Poll Register (IP)

The host begins the initialization sequence by either issuing a bus initialize or by using the IP initialize operation. Any write to that address will cause an initialization of the controller. When read while the controller is operating, it causes the controller to initiate polling. The WQESD responds to the 16 bit initialization words as set out in Table 4-4.

TABLE 4-4 WQESD INITIALISATION WORDS

NUMBER		FUNCTION	PROCESSOR
octal	hex		
000250	00A8	WOMBAT	PDP/11
000254	00AC	WOMBAT	MicroVAX II
000260	00B0	WOMBAT	On board port

#### Status and Address Register (SA)

The SA register consists of a set of two registers, the SA read register and the SA write register.

When read by the host during initialization, it communicates data and error information relating to the initialization process. When written by the host during initialization, it communicates certain host-specific parameters to the controller.

When read by the host during normal operation, it communicates status information including fatal errors detected by the controller.

#### 4.7 MSCP Commands

TABLE 4-5 WQESD MSCP COMMANDS

COMMAND	FUNCTION
Access	Reads data from the specified unit.
Abort	Guarantees that referenced MSCP command will complete within the controller timeout period.
Available	If specified unit is on-line, returns it to the unit-available state. If specified unit is currently in the unit-available state, this command has no affect.
Compare Host Data	Reads data from the disc and compares it with the data in the host buffer.
Erase	Writes zeros to the specified logical blocks on the unit. (No data is accessed from the host).
Get Command Status	Reports on the status of a specified command by returning a number that reflects the command's progress.
Get Unit Status	Reports on the status of a specified unit.
On Line	Places the specified unit on line, if possible.
Read	Reads data starting from the specified logical block on the disc, into host memory.
Set Controller Characteristics	Sets host-settable controller characteristics.
Set Unit Characteristics	Sets host-settable unit characteristics.
Write	Writes data starting at the specified logical block on the disc, from the host memory.

## 4.8 Error Handling

High data integrity is achieved by the controller through a 48 bit ECC (error checking and correction) polynomial with an 11 bit correction span. ECC will first try to read or write a block up to 10 times before attempting to correct the error. If error correction fails a non-recoverable error is reported. Table 4-7 details the MSCP status code messages.

## 4.9 Fatal Controller Error

If a fatal error is detected when the controller is initialized, the red error LED flashes and the fatal error status is set in the SA register. Table 4-6 describes fatal controller errors.

**TABLE 4-6 FATAL CONTROLLER ERRORS**

---

ERROR		DESCRIPTION
octal	hex	

---

125413	AB0B	RAM test failed
126013	AC0B	ROM test failed
126413	AD0B	Cache test failed
127013	AE0B	Cache unlock error
127413	AF0B	Qbus error - get host message
130013	B00B	Qbus error - clear host rings
130413	B10B	Buffers ; free buffer count error
131013	B20B	Queue empty - unfork
131413	B30B	Queue empty - wait
132013	B40B	Queue empty - pause
133013	B60B	Cache parity

---

TABLE 4-7    MSCP STATUS CODE MESSAGES

---

MESSAGE	MEANING
<hr/>	
Command Aborted	The current command was aborted before it could be completed normally.
Compare Error	While performing a Compare command, a Discrepancy was found while comparing the disc data to the host data.
Controller Error	The WQESD controller detected an internal error, but is able to continue processing its outstanding commands.
Data Error	An error was detected in the reading or writing of data. ECC attempts to read or write data up to 10 times. If the error persists correction is attempted. If correction fails the error is reported.
Drive Error	A drive-related error was detected (such as a seek failure).
Media Format Error	Indicates that the media mounted on the unit was incorrectly formatted.
Host Buffer Access Error	Reports bus timeouts and parity errors during data transfers. (Applies only to the data portion of an MSCP command).
Invalid Command	The WQESD controller found some field in the command to be in error.
Success	The command was successfully completed.
Unit Available	The WQESD controller is not on line, but it can accept an On Line command from the host.
Unit Offline	The WQESD controller is not on line, and it cannot be brought on line.
Write Protected	A Write or Erase command was attempted to a unit that is logically write-protected.

---

## Chapter 5 WQESD Operating Systems

The following discussion is intended to supplement DEC operating system resources and aims to aid the user of the WQESD in understanding how different operating systems integrate the device. This information will help the user of the controller plan the installation and in choosing the appropriate bus addresses and interrupt vectors for the the disc subsystem. For a complete description the DEC system documentation should be consulted.

### 5.1 Operating Systems Overview

In order to install any new device in a computer, the host operating system must be informed of the device's existence and where to find that device. In DEC operating systems this can be done in one of the following ways :-

- (a) The device can be manually connected using CONNECT or Configure statements.
- (b) The operating system can be informed about the peripheral device during an interactive SYSGEN.
- (c) The operating system can poll the device I/O address space.

Any of these methods will accomplish the desired result. The host system will be alerted to the device's existence, type, address and interrupt vectors.

Method (a) creates a command file that is executed on power-up. Method (b), interactive sysgen, creates a configuration file that the operating system accesses on power-up. Method (c) is referred to as 'autoconfigure'. RT-11 does not use autoconfigure but references standard bus addresses where it expects to find a device. All DEC operating systems try to follow the same set of rules but there are differences. These are discussed below.

### MSCP Devices

The Webster WQESD is an MSCP (Mass Storage Control Protocol) type device. All MSCP-type devices contain two registers that are visible to the Qbus I/O page. They are the Initialisation and Polling (IP) register and the Status and Address (SA) register.

### Qbus Addresses

The standard Qbus address of 17772150 (Octal) is used by all of the operating systems described in this manual as the address of the first controller on the host system. The IP register, CSR address, Qbus address and the base address all refer to the same register.

### Vector Addresses

Many operating systems choose vector addresses automatically. If an operating system requires manual input of vector addresses they are programmed into the controller during the initialisation process.

### Device Names

Table 5-1 is a list of device names for five operating systems. Two controller and device names are given to indicate the numbering scheme.

TABLE 5-1 DEVICE NAMES IN DEC OPERATING SYSTEMS

OPERATING	CONTROLLER		DRIVE	
	1st:	2nd:	1st:	2nd:
RSTS/E	RU0,	RU1	DU0,	DU1
RSX-11M	---	---	DU0,	DU1
RSX-11M-PLUS	DUA	DUB	DU0,	DU1
RT-11	Port0,	Port1	DU0,	DU1
VAX/VMS	PUA,	PUB	DUA0,	DUA1

## 5.2 RT-11 Operating System

### Installation of a Single Controller

A single controller is installed at the Qbus address of 17772150 (Octal) where RT-11 will find and then install the handler for that device. It is not necessary to run sysgen for a single controller. One of the pre-generated monitors provided with the RT-11 distribution kit can be used. To properly implement disc partitioning, the system start-up file (STARTx.COM) must be modified.

### Installation of Multiple Controllers

There are two valid methods that can be used to install multiple controllers. Either by modifying the MSCP handler, which is described in the RT-11 Software Support Manual or by performing a Sysgen. The following procedure describes the SYSGEN technique with user input marked in boldface type.

1. Initiate SYSGEN:

IND SYSGEN <return>

2. The system will then prompt the user by asking questions. The first concerns the use of a start-up command file when booting.

Do you want the start-up indirect  
file (Y)?    **Y**<return>

The start-up file performs two main functions. These specify the additional controller addresses and ensure that disc partitioning is carried out consistently on each bootstrap or power-up.

3. Select the device DU: as the MSCP device when prompted for Disc Options.

Enter the device name you want support for  
[dd]: **DU**<return>



4. Inform the system of the number of controllers to be installed.

How many ports are to be  
supported (1)? 2<return>

RT-11 refers to individual MSCP controllers on the host as ports. Each port has its own Qbus and vector addresses.

5. All other devices in the host computer configuration have to be specified. After completing this step, indicate that there are no more devices by entering a period (..).

Enter the device name you want support for  
[dd]: .<return>

6. Using the SET CSR keyboard command, specify the address of all the MSCP controllers. These must be added to the system start-up file STARTx.COM. The 'x' indicates the monitor to be used - S for single job, F for foreground/background, and X for extended memory. The command file must be edited to include the following statements :-

```
SET DU CSR0      = 17772150 (DEFAULT)
SET DU CSR1      = 17760334
SET DU VECTOR    = 154      (DEFAULT)
SET DU VEC2      = 160
```

The second device can be at any unused address on the Qbus I/O page supported by the pin settings on the controller. The vector address can be any unused address in the vector page. No default statements are required.

## Disc Partitioning Under RT11

Drives with capacities greater than 65,535 blocks (33.5 Mbytes) cannot be handled by RT-11 unless they are partitioned into smaller segments. Each partition can be smaller than 65,535 blocks if desired but there is a maximum of eight logical devices per physical drive. Each logical drive will be addressed by RT-11 as an independent physical drive.

The assignment names of each logical drive must be placed in the start-up command file to ensure that the drives are partitioned consistently and automatically each time the system is booted. The following is an outline of the procedure used to determine the number of logical drives to be assigned to each physical drive.

1. Decide on the drive configuration to be used. The logical unit number (LUN) and data storage capacity in logical blocks of each logical drive must be known. The controller plug settings must correspond to the bus address selected.
2. The total number of logical discs any physical disc can be partitioned into is calculated by dividing the selected block size of each logical disc into the total capacity of the the disc unit. Round the result to the nearest whole number. The last partition can be less than the maximum size selected. This number equals the number of logical discs.
3. STARTx.COM must now be edited to include the logical names of each partition. The format of each statement is :-

SET DUn UNIT=y PART=x PORT=z

where 'n' is the logical device name, 'y' is the unit number, 'x' is the partition number, and 'z' is the controller number. This must be done for each partition on each drive, including drives that have only one partition.

### Sample Disc Partitioning Procedure

The following is an example of the disc partitioning procedure for a drive of 245,412 blocks and a drive of 204,800 blocks. It has been decided to partition the drives into logical units of 65,535 blocks.

$$\begin{array}{lcl} \text{(a)} & \begin{array}{r} 245,412 \\ \hline 65,535 \end{array} & = 3.74 \text{ ( 4 logical units )} \\ \\ \text{(b)} & \begin{array}{r} 204,800 \\ \hline 65,535 \end{array} & = 3.12 \text{ ( 4 logical units )} \end{array}$$

Dividing the unit capacities by 65,535 and rounding the result to the nearest whole number gives the number of logical units. If the remainder is very small (under 800 blocks) then it would be advisable to round the figure down rather than up to the next highest number. This may avoid problems with partitions that are too small to be practicable.

Logical names can then be assigned to the partitions beginning with DU0 on controller unit 0 and modifying the start-up file to include the assignments.

```
SET DU0 UNIT=0 PART=0 PORT=0
SET DU1 UNIT=0 PART=1 PORT=0
SET DU2 UNIT=0 PART=2 PORT=0
SET DU3 UNIT=0 PART=3 PORT=0

SET DU4 UNIT=1 PART=0 PORT=0
SET DU5 UNIT=1 PART=1 PORT=0
SET DU6 UNIT=1 PART=2 PORT=0
SET DU7 UNIT=1 PART=2 PORT=0
```

### 5.3 RSTS/E Operating Systems (V8.0 and above)

RSTS/E can support two MSCP type controllers. The first is located at the standard Qbus address (17772150 octal) while the second can be located in floating address space. However, the recommended address for the second controller is 17760334. A controller must be located at the standard Qbus address to be a bootstrap device.

A program called INIT.SYS scans the system on power-up. INIT.SYS references a user-specified table located in the currently installed monitor. To alter the autoconfigure algorithm, the HARDWARE sub-option of INIT.SYS is used. This modifies the configuration table and allows an MSCP controller to be placed at any address on the I/O page. If a new monitor is installed then the table must be reset.

Controllers are assigned vector addresses and programmed by INIT.SYS during initialisation.

Warning: RSTS/E supports discs of a maximum size of 1,048,576 blocks. Larger drives must be broken up into multiple smaller virtual units. At a later date RSTS/E may support larger discs, refer to the RSTS/E Software Despatch for details.

#### 5.4 RSX-11M Operating Systems (V4.0 and above)

The RSX-11M SYSGEN program is an interactive program that builds a complete and running RSX-11M system for a particular hardware configuration. RSX-11M SYSGEN supports autoconfigure. This program detects MSCP type controllers located at standard Qbus addresses. Additional controllers must be manually attached to the system according to the procedure outlined below. The procedure is fully outlined in the RSX-11M System Generation and Configuration Guide.

##### Installing a single controller

A single controller is installed at the standard Qbus address of 17772150 (Octal). Autoconfigure can then be used to connect peripheral devices.

##### Installing multiple MSCP controllers

For two controllers manual initialisation must be undertaken. The following procedure will connect the devices to the operating system :-

1. Invoke SYSGEN.

```
>SET/UIC=[200,200]<return>
>SYSGEN<return>
```

2. Indicate that AUTOCONFIGURE has to be used by answering Y (Y) to the following :-

```
*   Autoconfigure the host system hardware?
[Y/N]: Y<return>
```

3. Indicate that the autoconfigure results are not to be overridden. Answer N (no) to the following :-

```
*   Do you want to override Autoconfigure
    results? [Y/N]: N<return>
```

Continue to answer the SET-UP questions as required then continue onto the TARGET CONFIGURATION section. Target configuration defaults for the first group of questions should be accurate because autoconfigure was used.

4. Indicate the number of devices that are installed.

- \* Devices: DU=2<return>
- \* Devices: .<return>

Enter the value the correct value of two. The period (.) terminates the device input operation.

The questions over the next four sections - HOST CONFIGURATION, EXECUTIVE OPTIONS, TERMINAL DRIVER OPTIONS, and SYSTEM OPTIONS - should be answered appropriately.

5. After answering the above sections it is necessary to define the PERIPHERAL OPTIONS for the controllers on the system. The questions will be asked once for each controller. The abbreviated form of controller "contr" is used.

The first prompt is for the interrupt vector address, Qbus address, the number of DU-type disc drives, the number of command rings, and the number of response rings. There is no default value for the number of disc drives.

- \* DU contr 0 [D:154,17772150,,4,4]
- \* 154,17772150,3,4,4<return>

#### Vector and Qbus Addresses

The standard vector address for MSCP controllers is 154 (octal). Any unused vector between 300 (octal) and 774 (octal) can be allocated for the second unit.

The standard Qbus address of 17772150 (octal) is used for the first controller, while the second can be 17760334 (octal) or in floating LSI-11 address space.

#### Drive Configuration

The following is a list of DEC manufactured drives that are DEC operating system compatible. Non-DEC drives must be compatible with those listed below.

If in doubt consult the manufacturer's specifications to verify compatibility.

- \*RX50
- \*RD51
- \*RD52
- \*RC25
- \*RA60
- \*RA80
- \*RA81

Count each RX50 drive as two drives, these contain two 5.25 inch floppy discettes. The RC25 has both fixed and removable media and should also be counted as two drives.

The configuration of the drives and the logical arrangement (disc partitions) for the disc sub-system is programmed by WOMBAT.

#### MSCP Ring Buffers

Command and response ring buffers which MSCP establishes in main memory also have to be specified. RSX-11M supports a maximum of eight rings. A value of four will minimise system overhead and is the recommended and default value.

6. The type of disc drives on each controller must now be specified.

\*DU contr 0 unit 0. is an RA60/80/81/RC25/RD51/rx50  
[D:RA81] RD51<return>

For the RQDX1, indicate that there is a RD51 and two RX50 drives. For the WQESD, indicate that there is one RD51 for each logical disc drive.

RSX-11M must have contiguous unit numbers which must be the same as those reported by the controller during initialisation.

Warning: Versions of RSX-11M prior to 4.2C support discs of a maximum size of 1,044,480 blocks. Larger drives must be broken up into multiple smaller virtual units.

## 5.5 RSX-11M-PLUS Operating Systems (V2.1 and above)

As with RSX-11M an interactive SYSGEN will build a complete running version of RSX-11M-Plus for a particular hardware configuration. RSX-11-Plus supports autoconfigure and will detect the first controller located at the standard Qbus address. Additional controllers must be installed manually.

### Installing a Single Controller

A single controller is installed at the standard Qbus address of 17772150 (octal) using autoconfigure to connect the peripherals. The procedure is fully outlined in the RSX-11M-Plus System Generation and Configuration Guide.

### Installing Multiple Controllers

To add the WQESD to the system configuration use the Add a Device option of SYSGEN or do a complete SYSGEN. The Add a Device procedure is described below :-

1. Invoke SYSGEN

```
>SET/UIC=[200,200]<return>
>@SYSGEN<return>
```

2. Answer **N** (no) to the following questions to indicate that only a subset of the SYSGEN procedure is wanted :-

- \* SU120 Do you want to do a complete SYSGEN?  
[Y/N D:Y]: **N**<return>
- \* SU130 Do you want to continue a previous SYSGEN  
from some point? [Y/N D:Y]: **N**<return>



3. Indicate that a specific module of SYSGEN is required by answering Y (yes) to the following :-

\* SU150 Do you want to do any individual sections of SYSGEN? [Y/N D:Y]: Y<return>

4. Select the Add a Device option of SYSGEN by typing the letter H.

\* SU160 Which sections would you like to do?  
[S R:0.-15.]: H<return>

SYSGEN now asks questions about the type and number of controllers to be installed in the system. There is one question for each controller supported. Type 0 (zero) until the prompt for UDA-type devices appears.

5. Specify the number of MSCP devices when asked by typing :-

\* CP3004 How many MSCP disc controllers do you have? [D R:0.-63. D:0.] 2<return>

6. Give the total number of drives on each controller installed on the system.

\* CP3008 How many MSCP disc drives do you have?  
[D R:0.-n. D:1.] 5<return>

The following is a list of DEC manufactured drives that are DEC operating compatible. Non-DEC drives must be compatible with those listed below. If in doubt consult the manufacturer's specifications to verify compatibility.

\*RX50  
\*RD51  
\*RD52  
\*RC25  
\*RA60  
\*RA80  
\*RA81

Count each RX50 drive as two drives, these contain two 5.25 inch floppy discs. The RC25 has both fixed and removable media and should also be counted as two drives.

The configuration of the drives and the logical arrangement (disc partitions) for the disc sub-system is programmed by WOMBAT.

7. SYSGEN then asks the user to specify controllers for each drive.

\* CP3044 To which DU controller is DU0:  
connected? [S R:1-1]: A<return>

This question is repeated until the number of MSCP drives has been exhausted. RSX-11M-Plus must have contiguous unit numbers and be the same as those reported by the controller during initialisation or errors will occur. Use A as the primary and B as the alternate controller.

8. Enter the Vector Address for each controller.

\* CP3068 Enter the vector address of DUA  
[O R:-774 D:154]

The standard vector address for MSCP controllers is 154 (octal). Any unused vector between 300 (octal) and 774 (octal) can be allocated for the second unit.

9. Enter the CSR address for each controller.

\* CP3076 What is its CSR address?  
[O R:1.-8. D:4.] 4<return>

The standard CSR address 17772150 (octal) is used for the first controller, while the second can be 17772154 (octal) or in floating CSR address space.

10. Specify the number of command rings for each MSCP controller.

\* CP3076 Enter the number of command rings for  
DUA [D R:1.-8. D:4.] 4<return>

RSX-11M-Plus supports a maximum of eight command rings. A value of four will minimise system overhead and is the recommended and default value.

11. Specify the number of response rings for each MSCP controller.

\* CP3076 Enter the number of response rings for  
DUA [D R:1.-8. D:4.] 4<return>

RSX-11M-Plus supports a maximum of eight response rings. A value of four will minimise system overhead and is the recommended and default value.

Warning: Versions of RSX--M Plus prior to 3.0C support discs of a maximum size of 1,044,480 blocks. Larger drives must be broken up into multiple smaller virtual units.

## 5.6 MicroVMS Operating Systems

The first WQESD controller is located at the standard bus address of 17772150 (Octal) and the second in floating address space. The MicroVMS SYSGEN utility can determine the Qbus and interrupt vector addresses for any of the I/O devices installed on the bus. MicroVAX/MicroVMS must be running in order to use this utility. The Qbus and interrupt vector addresses can be determined manually if access to a running system is not possible.

### Using MicroVAX/MicroVMS SYSGEN

The following is an outline of the MicroVMS SYSGEN procedure to determine Qbus and Interrupt vector addresses. This procedure requires system manager privileges.

1. Login and run the SYSGEN utility.

```
$ RUN SYS$SYSTEM:SYSGEN<return>
SYSGEN>
```

The SYSGEN> prompt indicates that the program is ready.

2. Obtain a list of the devices currently installed on the MicroVAX Qbus by typing :-

```
SYSGEN> SHOW/CONFIGURATION<return>
```

and get :-

```
Name: PUA Units: 1 Nexus: 0 CSR: 772150 Vector1: 154 Vector2: 000
Name: TXA Units: 1 Nexus: 0 CSR: 760500*Vector1: 310*Vector2: 000
```

\* Indicates a floating vector or address.

Sysgen lists the devices already installed on the Qbus by logical name. Devices with floating bus and vector addresses should be noted if it is intended to re-install them with the WQESD controller. Floating bus addresses will be larger than 760000 (octal). Floating interrupt vectors will be larger than 300 (octal).

3. Execute the configure command. This will determine the Qbus and Vector addresses that autoconfigure will expect for each device type.

```
SYSGEN> CONFIGURE<return>
DEVICE>
```

Specify the devices to be installed on the bus by typing their Qbus names. Under MicroVAX/MicroVMS the device name for MSCP-type controllers is UDA.

```
DEVICE> UDA,2<return>
DEVICE> DHV11<return>
```

The device name is separated from the number of devices by a comma. The number of devices is specified in decimal.

Devices with floating addresses or vectors are not affected by devices with fixed addresses or vectors. Only devices with floating addresses or vectors need be specified.

4. When all the devices have been specified enter a control-Z.

```
DEVICE> ^Z
```

The addresses and vectors of the devices entered will be listed in the following manner :-

```
Device: UDA    Name: PUA CSR: 772150 Vector: 154 Support: yes
Device: UDA    Name: PUB CSR: 760334* Vector: 300* Support: yes
Device: DHV11  Name: TXA CSR: 760500* Vector: 310* Support: yes
```

\* Denotes floating bus and interrupt vector addresses. Floating CSR addresses must be programmed into the WQESD by selecting the correct pin configuration on the PCB.

5. If an address other than that selected for the WQESD by CONFIGURE command is desired, CONNECT statements must be entered into the SYSCONIF.COM file. SYSCONIF.COM can only be accessed through the system manager's account SYSSMANAGER. The correct syntax is given in the DEC MicroVMS SYSGEN documentation.

The STARTUP.COM or UVSTART.COM command files in the main system account, SYSSSYSTEM must not be altered.

## 5.7 Autoconfigure

Autoconfigure is a utility program that finds and identifies I/O devices in the I/O page of system memory. Most devices have a fixed bus address reserved for them. When the computer is bootstrapped autoconfigure polls those addresses - specifically the console status register (CSR) which is usually the first register of the block.

A block of addresses is reserved when a device is detected. The size of the block is determined by the number of registers the device uses. Autoconfigure then looks to the next CSR address space for that same type of device. If there are no other devices of that type autoconfigure looks to the next valid CSR address. Autoconfigure expects an eight byte block to be reserved for each device not installed in the system. An empty block tells autoconfigure to look to the next valid address space.

Devices with no fixed address are assigned addresses from floating CSR address space. This may be necessary if there are several of the same device in the system. Floating address space is in the vicinity of 76000 to 763776 of the bus I/O page. Devices can also have floating interrupt vector addresses. Floating CSR and interrupt vectors must be assigned in specific sequences depending on the rank of the device (see Table 5-2 ). The presence or absence of floating bus and interrupt vector address devices will affect the assignment of addresses to other floating vector devices.

TABLE 5-2 SYSGEN DEVICE RANKING

Rank	Device	Number of Registers	Octal Modulus	Rank	Device	Number of Registers	Octal Modulus
1	DJ11	4	10	17	Reserved	4	10
2	DH11	8	20	18	RX11 <sup>2</sup>	4	10
3	DQ11	4	10	18	RX211 <sup>2</sup>	4	10
4	DU11, DUV11	4	10	18	RXV11 <sup>2</sup>	4	10
5	DUP11	4	10	18	RXV21 <sup>2</sup>	4	10
6	LK11A	4	10	19	DR11-W <sup>3</sup>	4	10
7	DMC11	4	10	20	DR11-B <sup>3</sup>	4	10
7	DMR11 <sup>1</sup>	4	10	21	DMP11	4	10
8	DZ11	4	10	22	DPV11	4	10
8	DZV11	4	10	23	ISB11	4	10
8	DZS11	4	10	24	DMV11 <sup>2</sup>	8	20
8	DZ32	4	10	25	DEUNA <sup>2</sup>	4	10
9	KMC11	4	10	26	UDA50 <sup>2</sup>	2	4
10	LPP11	4	10	27	DMF32	16	40
11	VMV21	4	10	28	KMS11	6	20
12	VMV31	8	20	29	VS100	8	20
13	DWR70 <sup>2</sup>	4	10	30	TU81	2	4
14	RL11 <sup>2</sup>	4	10	31	KMV11	8	20
14	RLV11 <sup>2</sup>	4	10	32	DHV11	8	20
15	LPA11-K <sup>2</sup>	8	20	33	DMZ32	16	40
16	KW11-C	4	10	34	CP132	16	40

- <sup>1</sup> DZ11-E and DZ11-F treated as two DZ11s.
- <sup>2</sup> The first device of this type has a fixed address while extra devices have floating addresses.
- <sup>3</sup> The first two devices of this type have fixed addresses while extra devices have floating addresses.

An eight byte gap must also be reserved in floating address space for each device type not currently installed in the system. This gap must start on the proper boundary. See Table 5.6 for an example of gap placement.

A device's CSR address is determined on word boundaries according to the number of bus accessible registers the device has. The relationship of word boundaries and device registers is set out in Table 5-3 Autoconfigure only inspects for a device type at one of the possible device boundaries. For instance, autoconfigure will not look for a DMZ32 which has 16 registers at an address that ends in 20.

**TABLE 5-3 DEVICE REGISTERS AND WORD BOUNDARIES**

Device Registers	Possible Boundaries
1	Any word
2	XXXXX0, XXXXX4
3, 4	XXXXX0
5, 6, 7, 8	XXXX00, XXXX20, XXXX40, XXXX60
9 thru 16	XXXX00, XXXX40

### Vector Addresses and Autoconfiguration

Devices are assigned vector addresses in order of rank commencing at 300 (octal) up to 777 (octal). Extra devices of the same type are assigned consecutive vector addresses according to the number of vectors required and starting boundaries for each device type. Table 5-4 shows the order of assignment.

The boundaries in the modulus column indicate where vector addresses are assigned. If the modulus is 10 the first vector address for that device must end with a zero (XX0). If the modulus is 4 the first vector must end with either a zero or four (XX0, XX4).

Vector addresses can only end on an address of four or zero i.e. modulo 4 boundaries (XX0, XX4). If a device has two vectors the first must start on a modulo 10 boundary. Using 350 as a starting point the vectors will be 350 and 354.



TABLE 5-4 FLOATING VECTOR ADDRESS DEVICE PRIORITY RANKING

Rank	Device	Number of Vectors	Octal Modulus
1	DC11	2	10
1	TU58 <sup>1</sup>	2	10
2	KL11 <sup>1</sup>	2	10
2	DL11-A <sup>1</sup>	2	10
2	DL11-B <sup>1</sup>	2	10
2	DLV11-J <sup>1</sup>	8	40
2	DLV11,DLV11-F <sup>1</sup>	2	10
3	DP11	2	10
4	DM11-A	2	10
5	DN11	1	4
6	DM11-BB/BA	1	4
7	DH11 modem control	1	4
8	DR11-A, DRV11-B	2	10
9	DR11-C, DRV11	2	10
10	PA611 (reader + punch)	4	20
11	LPD11	2	10
12	DT07	2	10
13	DX11	2	10
14	DL11-C TO DLV11-F	2	10
15	DJ11	2	10
16	DH11	2	10
17	VT40	4	20
17	VSV11	4	10
18	LPS11	6	40
19	DQ11	2	10
20	KW11-W, K WV11	2	10
21	DU11, DUV11	2	10
22	DUP11	2	10
23	DV11 + modem control	3	20
24	LK11-A	2	10
25	DWUN	2	10
26	DMC11	2	10
26	DMR11	2	10
27	DZ11/DZS11/DZV11	2	10
27	DZ32	2	10
28	KMC11	2	10
29	LPP11	2	10

continued on next page...

TABLE 5-4 FLOATING VECTOR ADDRESS PRIORITY RANKING

Rank	Device	Number of Vectors	Octal Modulus
30	VMV21	2	10
31	VMV31	2	10
32	VTV01	2	10
33	DWR70	2	10
34	RL11 <sup>2</sup> , RLV11 <sup>2</sup>	1	4
35	TS11 <sup>2</sup> , TU80 <sup>2</sup>	1	4
36	LPA11-K	2	10
37	IP11/IP300 <sup>2</sup>	1	4
38	KW11 <sup>2</sup> C	2	10
39	RX11 <sup>2</sup>	1	4
39	RX211 <sup>2</sup>	1	4
39	RXV11 <sup>2</sup>	1	4
39	RXV21 <sup>2</sup>	1	4
40	DR11-W <sup>2</sup>	1	4
41	DR11-B <sup>2</sup>	1	4
42	DMP11	2	10
43	DPV11 <sup>3</sup>	2	10
44	ML11	1	4
45	ISB11	2	10
46	DMV11 <sup>2</sup>	2	10
47	DEUNA <sup>2</sup>	1	4
48	UDA50 <sup>2</sup>	1	4
49	DMF32	8	40
50	KMS11	3	20
51	PCL11-B	2	10
52	VS100	1	4
53	Reserved	1	4
54	KMV11	2	10
55	Reserved	2	10
56	IEX	2	10
57	DHV11	2	10
58	DMZ32	6	20
59	CP132	6	20

- 1 KL11 or DL11 have fixed vectors when used as a console.
- 2 The first device has a fixed vector all subsequent device of the same type have a floating vector.
- 3 ML11 is a Mass Bus device which connects to the Qbus or Unibus via a bus adaptor.

### System Configuration Example

An example of a system configuration is shown in Table 5-5. The configuration includes both fixed and floating addresses and vectors.

TABLE 5-5 CSR AND VECTOR ADDRESS EXAMPLE

Controller	Vector	CSR
1 UDA50	154	772150
1 DZ11	300	760100
1 UDA50	310	760334
2 DHV11	320	760520
	330	760520

Table 5-6 shows the computed CSR addresses and gaps for floating devices.

TABLE 5-6 FLOATING CSR ADDRESS ASSIGNMENT

Installed	Device		Octal Address
	DJ11	Gap	760010
	DH11	Gap	760020
	DQ11	Gap	760030
	DU11	Gap	760040
	DUP11	Gap	760050
	LK11A	Gap	760060
	DMC11	Gap	760070
----->	DZ11		760100
		Gap	760110
	KMC11	Gap	760120
	LPP11	Gap	760130
	VMV21	Gap	760140
	VMV31	Gap	760150
	DWR70	Gap	760170
	RL11	Gap	760200
	LPA11-K	Gap	760220
	KW11-C	Gap	760230
	Reserved	Gap	760240
	RX11	Gap	760250
	DR11-W	Gap	760260
	DR11-B	Gap	760270
	DMP11	Gap	760300
	DPV11	Gap	760310
	ISB11	Gap	760320
	DMV11	Gap	760330
	DEUNA	Gap	760340
----->	UDA50 (WQESD)		772150 <sup>1</sup>
----->	UDA50 (WQESD)		760354
		Gap	760360
	DMF32	Gap	760400
		Gap	760440
	KMS11	Gap	760420
	VS100	Gap	761440
	TU81	Gap	761450
	KMV11	Gap	761460
----->	DHV11		761500
----->	DHV11		761520
		Gap	761530
	DMZ32	Gap	761540
	CP132	Gap	761600

<sup>1</sup> indicates a fixed address device

## 5.8 Error Logging

Firmware version 2.8H and later supports error logging so that various drive and data errors will be reported to the host.

The table below lists the return codes and their meaning.

### ERROR CODE CLASSES

The return codes are spilt into two main groups, Informational and Error.

#### Informational Error Codes

Informational entries note disk retries and disk blocks that have been transparently replaced. No action on the part of the user is required, although a block that has a high number of retries is a sign that it may be going bad.

#### Error Entries

Error entries indicate that something has been found that the firmware cannot handle or has incompletely dealt with. The error classes are:

a. Block Replacement Succeeded - Data Bad.

This occurs when the controller was unable to recover the data, but a block replacement occurred successfully. When this block is later written to, the 'forced error' flag associated with the block will be cleared.

b. Block Replacement Failed - Data Bad.

This occurs when the controller was unable to recover the data, and also unable to replace the block (probably means the RCT is full so a replacement block was unavailable).

c. Read of Block with Forced Error Flag.

This occurs when a block already replaced with bad data (as in point a.) is read again without having been written to since the block was replaced.

## ERROR CODE VALUES

The following values are returned for the two types of error message returned.

### Informational Error Codes

#### DATA ERROR

##### 1 Symbol ECC error

Level	Retry	
0	n	soft error, retried. n = of retries needed.
1	n	replace block, good data

#### Error Entries

#### DATA ERROR

##### 4 Symbol ECC error

Level	Retry	
0	0	reread of a replaced block with bad data
1	n	replace block, bad data (not recovered)
2	0	replace good failed but we got the data ok (read only)

#### DATA ERROR

##### Uncorrectable ECC error

Level	Retry	
0	ic	Bad block/catch all for errors, ic = Internal error Code This includes the case where the data is bad and the block replacement failed as well

## ERROR CODE OUTPUT

How the entries appear in the errorlog will differ depending on machine and operating system used but as an example I will use the MicroVAX II running VMS 4.7A.

Typing "ANALYSE/ERROR" will give a report similar to this.

```
=====
.
.
MSLG$W_UNIT          0003
                     UNIT  3.
.
.
MSLG$B_FORMAT        02
                     DISK TRANSFER ERROR
MSLG$W_EVENT         0108
                     DATA ERROR
                     1. SYMBOL ECC ERROR
.
.
(various other messages)
.

MSLG$B_UNIT_SVR      02
                     UNIT SOFTWARE VERSION  2.
MSLG$B_UNIT_HVR      08
                     UNIT HARDWARE REVISION  8.
MSLG$B_LEVEL         00
MSLG$B_RETRY         01
MSLG$L_VOL_SER       00000000
                     VOLUME SERIAL  0.
MSLG$L_HDR_CODE      00031764
                     LOGICAL BLOCK  202596
                     GOOD LOGICAL SECTOR
=====
```

In this example the error log is reporting an informational entry, as indicated by the "1.SYMBOL ECC ERROR" message. The firmware in use on the controller is 2.8 as given by the "UNIT SOFTWARE VERSION 2." and the "UNIT HARDWARE REVISION 8.". The "MSLG\$B\_LEVEL 00" is telling us that the message is about the need to retry a read to get data from the disk. The number of retries needed to achieve this is given in "MSLG\$B\_RETRY 01", in this case only one retry was needed. The disk block that had the error is reported last "LOGICAL BLOCK 202596" in this example. All this occurred on "UNIT 3.".

## Chapter 6 WQESD Cache Operation

### 6.1 WQESD Disc Cache

The WQESD implements a disc cache which is designed to facilitate larger and faster data transfers between disc and the host by reducing the time wasted on positioner operations. Even with fast drives 98% of disc time for continuous random access to single sectors of data is taken by positioner operations. The WQESD offers at least an 80% improvement in access times by reducing the number of disc accesses required. The WQESD cache is implemented as one megabyte of dynamic RAM.

### 6.2 Read Look-ahead

The WQESD allows the user to program the controller to perform read look-ahead in anticipation of impending data requests. The optimum look-ahead value can only be determined within system and application parameters but can range from 0 to 255 blocks. A value of zero will disable the feature. The default value is four.

The anticipated hit ratio for the WQESD cache is 90% although this can be reduced depending upon the nature of the data accessed. Because most user programs write and read data sequentially there is a high probability that in one fetch operation the controller will be able to satisfy several sequential data reads without the need for further disc accesses.

The cache has been designed to maximise the probability of finding the target data over a range of sequential and non-sequential reference patterns while minimising cache misses and controller overhead.

### 6.3 Cache allocation

Cache memory is used to hold the disc cache blocks, a cache map and fixed buffers for special usage. Data from the disc or main memory is stored in blocks at addresses determined by the cache assignment algorithm. Their contents and location are recorded in the cache map.

The cache map consists of a 4-byte entry for each cache block. The cache map is indexed by the cache block number and contains the address (drive number, logical block) of the current occupant together with flags (locked, valid, primary copy not written, shadow copy not written).



Fixed buffers are assigned for RCT buffers (1 per drive) and a single block buffer for disc management I/O. The location and size of all cache variables are held in RAM.

#### 6.4 Cache usage

All disc I/O is done via the cache. A set of cache blocks must be assigned for all transfers and continuous disc operations must be done via contiguous cache blocks. Disc and Qbus transfers are performed simultaneously.

#### 6.5 Cache Assignment Algorithm

The WQESD cache implements a contention based hashing algorithm to determine block replacement. A given disc block has a fixed cache address calculated as follows :-

- (a) Get the remainder of the logical block number modulo the number of cache blocks.
- (b) Bias this by a fixed offset which is a function of the drive number. (This is so that the same logical block on 2 discs have a different cache block number).

A disc block also has an alternative cache address calculated by biasing it by approximately half the number of cache blocks. The alternate cache block is only used for compare operations.

#### 6.6 Cache Operation

The following describes the cache operation algorithm :-

Read:

Examine cache for data required.

If all data in cache  
Transfer data from cache to Qbus

else

Assign cache (lock it and wait if locked already)  
Perform read  
Unlock cache and flag as valid

Write:

Assign cache (lock it and wait if locked already)  
Transfer from Qbus to cache and flag as valid

Perform write  
Unlock cache

## 6.7 Cache Disable

The WQESD cache can be disabled for performance evaluation, engineering, and diagnostic application requirements by selecting that option in WOMBAT. The cache cannot be selectively disabled for a particular drive.

## 6.8 Early Write Notification

The WQESD implements early write notification where data to be written to disc is retained in the cache and the host is issued a write complete notification. The controller will then write the data to the disc at the most convenient time.

It should be noted that in the event of system failure any data residing in the cache will be lost. The early write notification can be disabled by invoking WOMBAT and selecting the appropriate option.

## Chapter 7 WQESD Qbus Interface

### 7.1 Qbus Interface

All data, address and control information transfers between the processor and disc controller are carried out over the Qbus. The WQESD supports all current Qbus functions including block mode DMA, 22 bit addressing, 4-level position independent interrupt structure, all LSI-11 CPU's and MicroVAX II.

The Qbus consists of 42 bidirectional and 2 unidirectional signal lines wired into the backplane assembly. These are grouped into the following categories :-

- \* Sixteen multiplexed data/address lines - BDAL<15:00>
- \* Two multiplexed address/parity lines - BDAL<17:16>
- \* Four extended address lines - BDAL<21:18>
- \* Six data transfer control lines - BBS7L, BDINL, BDOUTL, BRPLYL, BSYNCL, BWTBTL
- \* Six system control lines - BHALTL, BREFL, BEVNTL, BINITL, BDCOKL, BPOKL
- \* Ten interrupt control and direct memory access control lines - BIAKOL, BIAKIL, BIRQ4L, BIRQ5L, BIRQ6L, BIRQ7L, BDMGOL, BDMRL, BSACKL, BDMGIL

Communication is asynchronous, allowing devices with differing data rates to share the bus. A strict master/slave protocol avoids the need for synchronising clock pulses by implementing handshaking and other control signals between I/O devices.

### 7.2 Interrupts

Interrupt priority for the WQESD is switch selectable on the PCB. The recommended priority setting is level four. Note that if level five is set and a third party controller is installed also set to level five, then a fatal BUGCHECK can occur.

Interrupts suspend program execution while the processor starts the device service routine at a vector address input from the requesting device.

Interrupts are serviced according to device priority. Device

priority can be determined in two ways. These are termed 'Position Defined' and 'Distributed' arbitration. Positioned Defined arbitration gives priority to those devices which are electrically closest to the processor. Distributed arbitration implements priority according to the priority levels set on the device hardware. When devices with equal priority generate an interrupt, the processor gives preference to the device which is electrically closest. A previous bus transaction must have been completed before another can be commenced.

The interrupt protocol has three phases :-

1. Interrupt Request Phase. The interrupt enable bit in the status register is set and interrupt request lines are asserted according to priority settings.
2. Interrupt Acknowledge and Priority Arbitration Phase. The processor detects the request and checks if any other device with higher priority is requesting an interrupt. If there are no devices with higher priority seeking an interrupt the processor acknowledges the interrupt.
3. Interrupt Vector Transfer Phase. The device outputs vector address bits to the processor which then enters the device service routine.

### 7.3 Direct Memory Access

The WQESD supports both normal and block mode Direct Memory Access (DMA). During a DMA transfer the processor passes mastership of the bus to the controller.

During block mode DMA transfer the WQESD has a four microsecond delay after every 16 words to service any pending interrupt or DMA requests from other devices. The WQESD also detects DMA requests from other devices and will implement a 'DMA Throttle' after eight words. This prevents data loss from other DMA devices which may also share the Qbus.

The WQESD interleaves address references with bursts of data during DMA. Because the starting memory address is asserted only once every sixteen data words so data throughput is almost doubled.

DMA protocol consists of three phases :-

1. Bus Mastership Acquisition Phase. The WQESD requests control of the bus. The processor arbitrates the request then initiates the transfer of bus mastership.

2. Data Transfer Phase. The processor provides the controller with the following information utilising MSCP - block number on the disc, the number of bytes to transfer, and address in main memory, and if the operation is a read or write.
3. Bus Mastership Relinquish Phase. Bus mastership is relinquished after completing or aborting the data transfer cycle.

For a detailed description of the Qbus the appropriate DEC manual should be consulted.

## Chapter 8 WQESD ESDI Interface

### 8.1 ESDI Interface

The Enhanced Small Device Interface is an industry standard developed by the ESDI Committee to provide a low cost interface suitable for smaller high performance memory devices. The ESDI interface consists of a 34 way control cable and a 20 way data cable.

### 8.2 Control and Data Cables

The control cable allows for a daisy chain connection of up to four drives with the last drive being terminated. The data cable must be attached in radial fashion. The maximum cable length is 3 metres (9.8 feet). Tables 8-1 and 8-2 provide control and data cable details.

**TABLE 8-1 ESDI CONTROL CABLE (J1/P1) SIGNALS**

SIGNAL NAME	SIGNAL PIN	GROUND PIN
-HEAD SELECT 2 <sup>3</sup>	2	1
-HEAD SELECT 2 <sup>2</sup>	4	3
-WRITE GATE	6	5
-CONFIG/-STATUS DATA	8	7
-TRANSFER ACK	10	9
-ATTENTION	12	11
-HEAD SELECT 2 <sup>0</sup>	14	13
-SECTOR/-BYTE CLOCK/-ADDRESS MARK FOUND	16	15
-HEAD SELECT 2 <sup>1*</sup>	18	17
-INDEX	20	19
-READY	22	21
-TRANSFER REQ	24	23
-DRIVE SELECT 1	26	25
-DRIVE SELECT 2	28	27
-DRIVE SELECT 3	30	29
-READ GATE	32	31
-COMMAND DATA	34	33

\* Unused but terminated or held false. Ground all odd pins

TABLE 8-2 ESDI INTERFACE DATA CABLE (J2/P2) SIGNALS

SIGNAL NAME	SIGNAL PIN	GROUND PIN
-DRIVE SELECTED	1	
-SECTOR/-BYTE CLOCK/-ADDRESS MARK FOUND	2	
-SEEK COMPLETE	3	
-ADDRESS MARK ENABLE	4	
-RESERVED FOR STEP MODE	5	6
+WRITE CLOCK	7	
-WRITE CLOCK	8	
-CARTRIDGE CHANGED	9	
+READ REF CLOCK	10	12
-READ REF CLOCK	11	12
+NRZ WRITE DATA	13	15
-NRZ WRITE DATA	14	16
+NRZ READ DATA	17	19
-NRZ READ DATA	18	19
-INDEX	20	

The four head select lines  $2^0 - 3$  allow selection of each individual read/write head with HEAD SELECT 0 being the least significant line. Heads are numbered 0 through 15. Head 0 is selected for removable media drives, Head 0 being selected when all HEAD SELECT lines are high (inactive). The SELECT HEAD GROUP command allows for the addressing of more than 16 heads. Head addressing is continuous from 0. Data can be written to or read from the disc when the WRITE GATE or READ GATE signals are active (low).

## 8.2 Command Data

COMMAND DATA consists of 16 information bits of serial data plus parity. The command data word structure is set out in Table 8-3 below. Flow control is through the handshake signals TRANSFER REQ and TRANSFER ACK with the MSB being transmitted first. The parity bit is odd.

TABLE 8-3 COMMAND DATA WORD STRUCTURE

MOST SIGNIFICANT BIT																LEAST SIGNIFICANT BIT			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	P			
CMD FUNCTION				CMD MODIFIER				ALL ZEROS								P			
CMD FUNCTION				CMD PARAMETER												P			

BIT P: PARITY (ODD)

Command Data bits 15 through 12 in combination with the Command Modifier bits 11 through 8 define a variety of ESDI functions. The Command Modifier bits are appended to each of the command functions where applicable. Table 8-4 shows the meaning of the various bit combinations. Detailed explanations of these functions can be found in the ESDI Interface Standard documentation.



TABLE 8-4 COMMAND (CMD) DATA DEFINITION

CMD FUNCTION BIT 15 14 13 12				CMD FUNCTION DEFINITION	CMD MODIFIER APPLICABLE BITS 11-8	CMD PARAMETER APPLICABLE BITS 11-0	STATUS CONFIG. DATA RETURNED TO CONTROLLER
0	0	0	0	SEEK	NO	YES	NO
0	0	0	1	RECALIBRATE	NO	NO	NO
0	0	1	0	REQUEST STATUS	YES	NO	YES
0	0	1	1	REQUEST CONFIG	YES	NO	YES
0	1	0	0	SELECT HEAD	NO	YES	NO
				GROUP (OPTIONAL)			
0	1	0	1	CONTROL	YES	NO	NO
0	1	1	0	DATA STROBE	YES	NO	NO
				OFFSET			
0	1	1	1	TRACK OFFSET	YES	NO	NO
1	0	0	0	INITIATE	NO	NO	NO
				DIAGNOSTICS			
				(OPTIONAL)			
1	0	0	1	SET BYTES PER	NO	YES	NO
				SECTOR			
				(OPTIONAL)			
1	0	1	0	RESERVED	-	-	-
1	0	1	1	RESERVED	-	-	-
1	1	0	0	RESERVED	-	-	-
1	0	0	1	RESERVED	-	-	-
1	1	1	0	RESERVED	-	-	-
1	1	1	1	RESERVED	-	-	-

## Chapter 9 WQESD Hardware Description

### 9.1 General Description

There are four major functional sections in the WQESD :-

1. The microprogrammed sequencer combined with an 8-bit RALU, forms a high speed processor which performs the following functions :-
  1. Implement the MSCP register pair - the Initialization and Polling Register and the Status and Address Register. Respond to accesses of these registers by the host CPU.
  2. When the Auto Boot option is enabled, respond to accesses of the first eight Boot locations (17773000 through 17773010 or 17771000 through 17771010) and initiate the automatic bootstrap function.
  3. Generate Qbus interrupt requests and control the vector transfer phase of the interrupt cycle.
  4. Generate Qbus DMA requests and control the transfer of data between cache memory and Qbus main memory. Use block mode DMA transfers if the main memory will support them, otherwise use conventional burst mode DMA. During DMA transfers, detect memory parity and timeout errors.
  5. During disc read/write operations, control the transfer of data between the disc FIFO buffer, (contained within the 8466 controller chip) and the cache memory.
  6. Implement a microprocessor-like instruction and register set. Fetch and execute these instructions from an on-board EPROM. This EPROM contains firmware modules which provide an ODT style debug, a set of disc formatting, testing and bad block management functions, and the high level MSCP protocol implementation.
2. The Qbus interface, which contains the necessary logic to support Data In (DIN), Data Out (DOUT), and Read Modify Write (DATIO) bus cycles from the host CPU. It also supports Interrupt Request cycles, DMA Request cycles and DMA transfer cycles with 22-bit addressing in both block mode and burst mode protocols.
3. The ESDI disc interface. Disc read, write and formatting functions are performed by the 8466 controller chip. Seek, head select, and drive select functions are performed by extra logic.

4. The cache memory. The WQESD uses 36 256K rams to implement 1Mbyte of storage for use as a disc cache. This memory is parity protected at the byte level. Circuitry is provided to refresh the memory every 3.3 mS.

## 9.2 Detailed Description

### Microprogrammed Sequencer

The sequencer consists mainly of a set of eight 1K registered PROMS which store a set of microcode routines, each of which performs a specific function. Execution of a microcode routine is initiated by the P PROM at Y5. The inputs of the P PROM are connected to various system states such that it can recognise when a particular routine should be executed.

The outputs of the P PROM are connected to the Next Address bus of the sequencer, uA1-uA8, so that when the appropriate input conditions arise the P PROM will assert the starting address of the desired routine to be executed. On the next system clock edge (SYSCLK), the P PROM will be disabled and the Next Address PROM (N PROM) will be enabled by the TRAP signal. The N PROM at X5 has its inputs and outputs connected to the Next Address bus, allowing it to use the current address on this bus to determine what the next address should be. This new address is placed on the Next Address bus on the next SYSCLK edge, thus entering the second step of the routine. On each SYSCLK edge, the N PROM provides the address of each successive step of the routine being executed. At the end of the routine the TRAP signal is asserted, returning control of the Next Address bus to the P PROM.

The least significant bit of the Next Address bus, uA0, can be made to assume the state of any one of the 22 system signals connected to the three 8-input multiplexers at W4, W6, Y4. The sequencer controls these multiplexers via the B PROM at W5, so that at any step in a routine the next address will be conditionally odd or even depending on the state of the selected input. This mechanism allows the sequencer to make decisions based on various system states, and to take the appropriate action. Seven other PROMS - the A,F,S,C,M,L, and E PROMS - are connected to the Next Address bus, each producing eight output signals which control the rest of the logic on the board. One of these signals, uA9, from the C PROM forms the most significant bit of the Next Address bus, and can be viewed as an extension of the N PROM.

If there are no pending requests at the input of the P PROM the sequencer enters an instruction fetch routine. This routine places the contents of a program counter, stored in the RALU, into the address latch made up of the five 74LS191 counters at H5, K6, H6,

J6, G6. The output of this latch addresses the U.V. erasable firmware PROM (EPROM) at K5. The sequencer then asserts the uREAD signal, followed by the FETCH signal, which places the output of the EPROM onto the Next Address bus via the octal latch at S4. This starts up a microcode routine which performs the function specified by the value fetched from the EPROM; thus executing the instruction.

The 8-bit Register and Arithmetic Logic Unit (RALU) made up of the two 2901C bit slice chips at N4, R4 performs all data manipulation functions and stores the "microprocessor" register set.

## Qbus Interface

Data and address information is transferred between the WQESD and the Qbus via the 2908 bus transceivers M8, N8, P8, R8. The 2908s at P6 and S8 drive the high order address lines during Qbus DMA, giving full 22-bit addressing capability. Incoming 16-bit data appear on DAL0-DAL15, and are multiplexed onto the internal 16 bit data bus (DB0 - DB15) via the 81LS95 octal buffers at L7, L8.

Incoming addresses also appear on DAL0 - DAL15 and are decoded by the WPROM at M6. This PROM produces eight outputs, six of which correspond to the six base address options of the WQESD. The other two outputs provide the two bootstrap address options (17773000, 17771000). The address and boot option links select the appropriate outputs from the W PROM to be input to the P PROM via address latch X6 and synchronising latch Y6. These inputs along with DAL1, and RDIN provide the P PROM with sufficient information to initiate the appropriate microcode routine to service the Qbus request.

16-bit data to be transferred to the Qbus are loaded into latches in the 2908 bus transceivers from the 16-bit data bus DB0 - DB15, under the control of microcode signals LDBRL/ and LDBRM/. Outgoing addresses are loaded in a similar manner, with the high order 6 bits being loaded by LDBRH/. When the bus transceivers have been loaded with the appropriate information, the bus drivers are enabled by signals BUSEN/ and BUSENH/.

To generate an interrupt request to the host CPU, the microcode asserts DEVIRQ/. This asserts BIRQ4L on the Qbus, and depending on the configuration of the interrupt priority links P1, P2, P3, may also assert BIRQ5L, BIRQ6L, and/or BIRQ7L. The host responds with BDINL followed by BIACKIL, which, provided that no higher priority device is requesting an interrupt, will cause the INTR signal at pin 6 of W7 to be asserted. This connects to the P PROM via Y6 and causes the interrupt vector transfer microcode routine to be executed.

The microcode initiates DMA transfers by asserting the DMR signal which asserts BDMRL on the Qbus. The host responds with BDMGIL which asserts the DMAGR signal on pin 6 of V7. This signal is connected to

the P PROM and initiates the DMA transfer routine in the microcode. This routine asserts TSACK to become Qbus master, then proceeds to transfer data over the Qbus using the block mode protocol, provided this is supported by the main memory as indicated by the BREFL signal. Otherwise, burst mode DMA is used.

## ESDI Interface

The 8466 disc controller chip performs the basic read, write and formatting functions required of the WQESD. During a read, it accepts serial NRZ data from the disc, converts it to 16 bit words and stores them in an on-chip 16 word FIFO for retrieval by the external logic. Conversely, during a write, it removes 16 bit words from the FIFO and converts them to serial NRZ data to be written to the disc. The chip also performs sector header recognition, CRC generation and checking (for headers) and ECC generation and checking (for data). The chip contains 64 internal registers which are made accessible to the WQESD firmware, and allow programming of disc format parameters, ECC polynomials, and disc commands.

The ESDI interface consists of a 20 way data cable (one per drive), over which serial disc data and clock information is transmitted, and a 34 way control cable which carries disc command and status information.

16 bit command information is transmitted serially to the drive on the COMMAND DATA line at R1 pin 8. 16 bit configuration and status information is received from the drive on the CONFIG/STATUS line at T1 pin 8. These 16 bit transfers are controlled by a simple handshake protocol involving the TRANSFER REQ signal, generated by the controller, and the TRANSFER ACK signal, generated by the drive. Refer to the ESDI interface specification for a complete description of this protocol.

The 8 bit addressable latch at M3 provides 4 head select and 3 drive select signals. The head select signals are binary coded and allow selection of any one of up to 16 heads. The drive select signals are also binary coded but note that a code of zero means deselect all drives. (There is no drive 0 in the ESDI specification.)

The WRITE GATE signal is generated by the 8466 controller chip and transmitted directly to the drive via the 7438 bus driver at R1. The RGATE signal from the 8466 is delayed by eighteen byte times before being transmitted to the drive as READ GATE. The delay is provided by the two counters at J4, K4, and ensures that READ GATE will never be asserted during a write splice.

NRZ disc data and clock signals are communicated to and from the drive via differential TTL lines for high noise immunity. The 3486 chips at H1, K1, receive the differential READ DATA and READ CLOCK signals from up to four drives, and the 26LS31 chips at J1, L1,

transmit differential WRITE DATA and WRITE CLOCK signals to the drive(s).

All received differential signals are terminated by 100 ohm resistors. All other received signals are terminated by 220 ohms to 5 volts and 330 ohms to ground provided by S.I.P resistor packs.

### Cache Memory

An array of thirty six 256K dynamic RAM chips forms a one megabyte parity protected memory which is used as a disc cache. The array is arranged as two banks of 18 chips with 128K 16-bit words (and two parity bits) in each bank. The array is addressed by the same address latch that addresses the firmware PROM mentioned above. 18 of these address bits are multiplexed by G5, F5, and E2 to form XA0 - XA8 which become the row and column address inputs for the array. Row addresses are strobed into the array by DRAS0/ or DRAS1/ depending on the state of the high order address bit A18. This provides a bank select function. Column addresses are strobed by DCASHI/ and DCASLO/. These two signals are used together to access a word, or individually to access the high or low byte only.

There is a parity bit assigned for each byte of data in the array. During a write cycle, parity information is generated by the two 74S280s at K7, K8, and written into the parity chips E5, E7, (low bank) or E6, E8, (high bank). Odd parity is used. During a read cycle, the same chips, K7, K8, check the parity bits and assert CPERR if an error is detected. If the WWP/ signal is asserted, wrong parity will be written into the array on all subsequent write cycles. This provides a diagnostic aid for testing the parity logic.

Dynamic RAMs require that 256 row addresses be refreshed every 4 mS. The 14 bit divider chip at V4 generates a refresh request every 12.8 uS by causing the REFR signal to be asserted at pin 6 of T3.0. This causes the sequencer to refresh one row address which is derived from the low order 8 bits of V4. By refreshing a row address every 12.8 uS, all 256 row addresses are refreshed in 3.3 mS.

### I/O Port

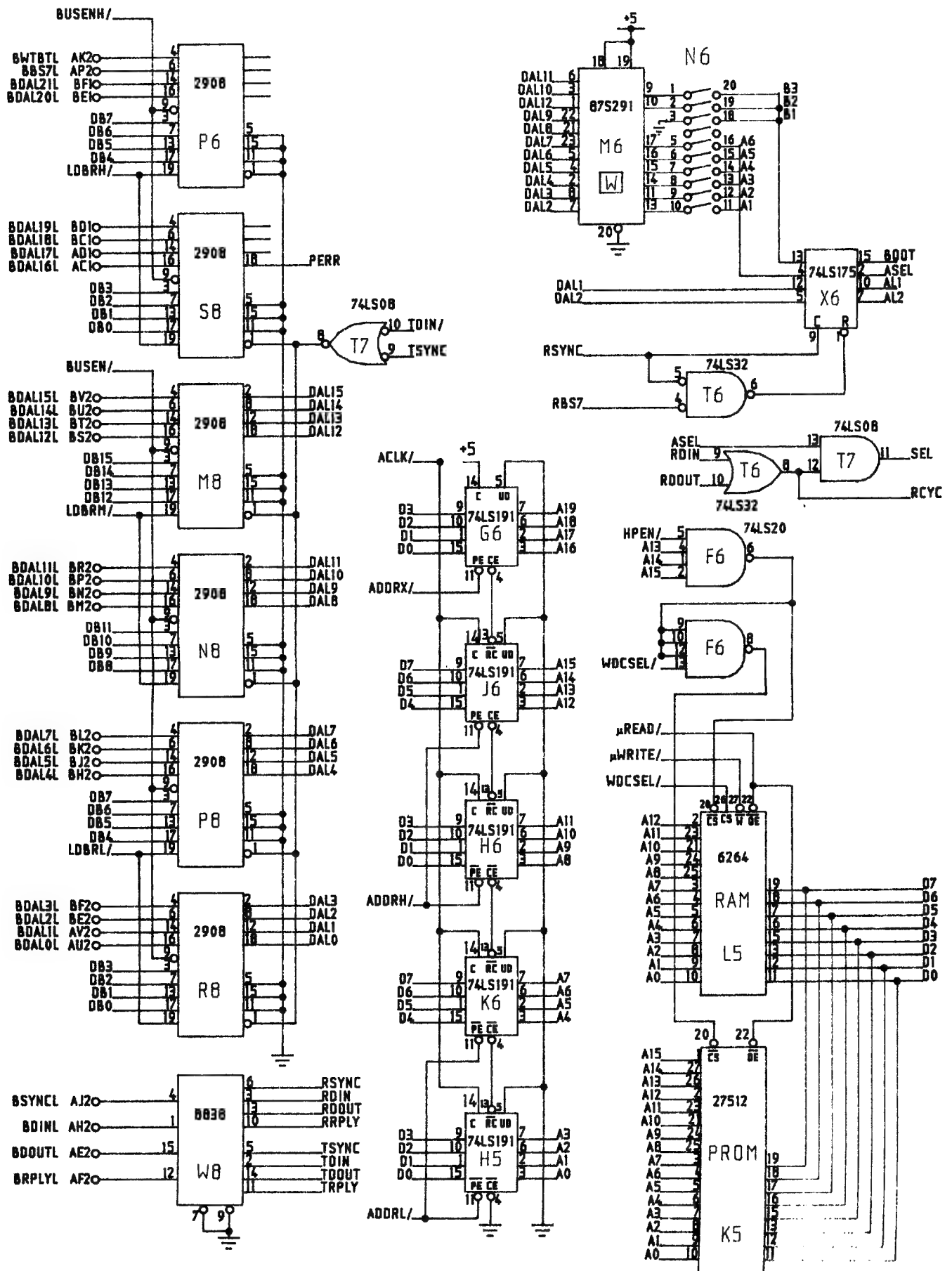
The I/O port serves two main functions. When running WOMBAT it can serve as an RS232 port to an ASCII terminal. This function is enabled by grounding the RS232 ENABLE/ input. If this input is left open then WOMBAT will try to communicate with the system console via a co-operating program in the host. The SERIAL signal from S3 pin 5 drives the base of the 2N4356 transistor, which produces the necessary +-5V swing at its collector to drive the RS232 output. The RS232 IN signal is clipped by a zener diode and then received by the octal buffer at V2.

The two TTL signals ACCESS1/ and ACCESS2/ indicate disc activity on drive 1 and 2 respectively. These signals are connected to the two on-board green LEDs and can be used to operate remote indicator lamps via appropriate driver circuits. The four signals WRITE PROTECT1/, 2/, 3/ and 4/ can be externally grounded to write protect drive 1, 2, 3 and 4 respectively.

## Chapter 10 WQESD Circuit Diagrams

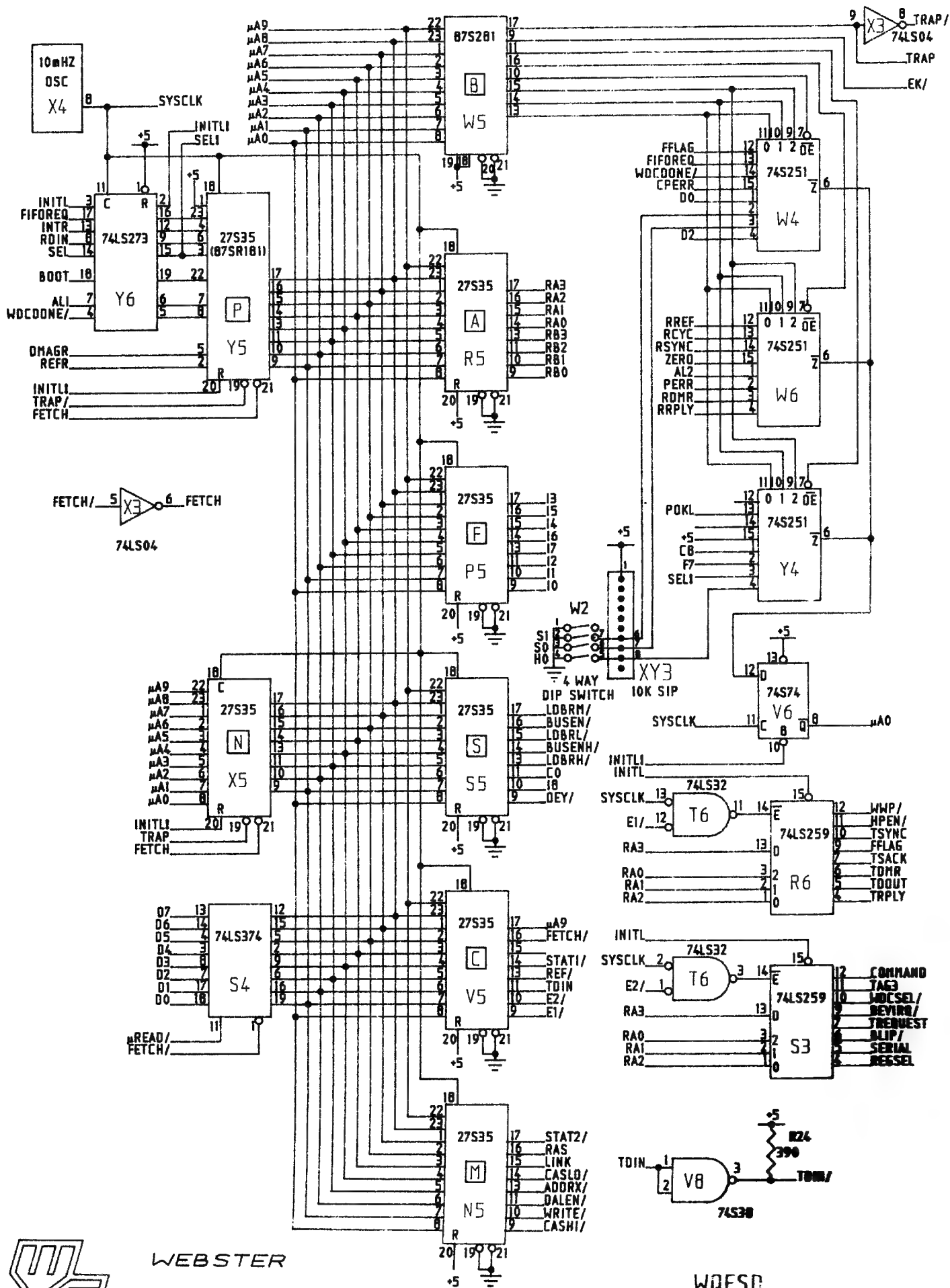
The circuit diagrams for the WQESD Disc Controller may be found in the following pages.





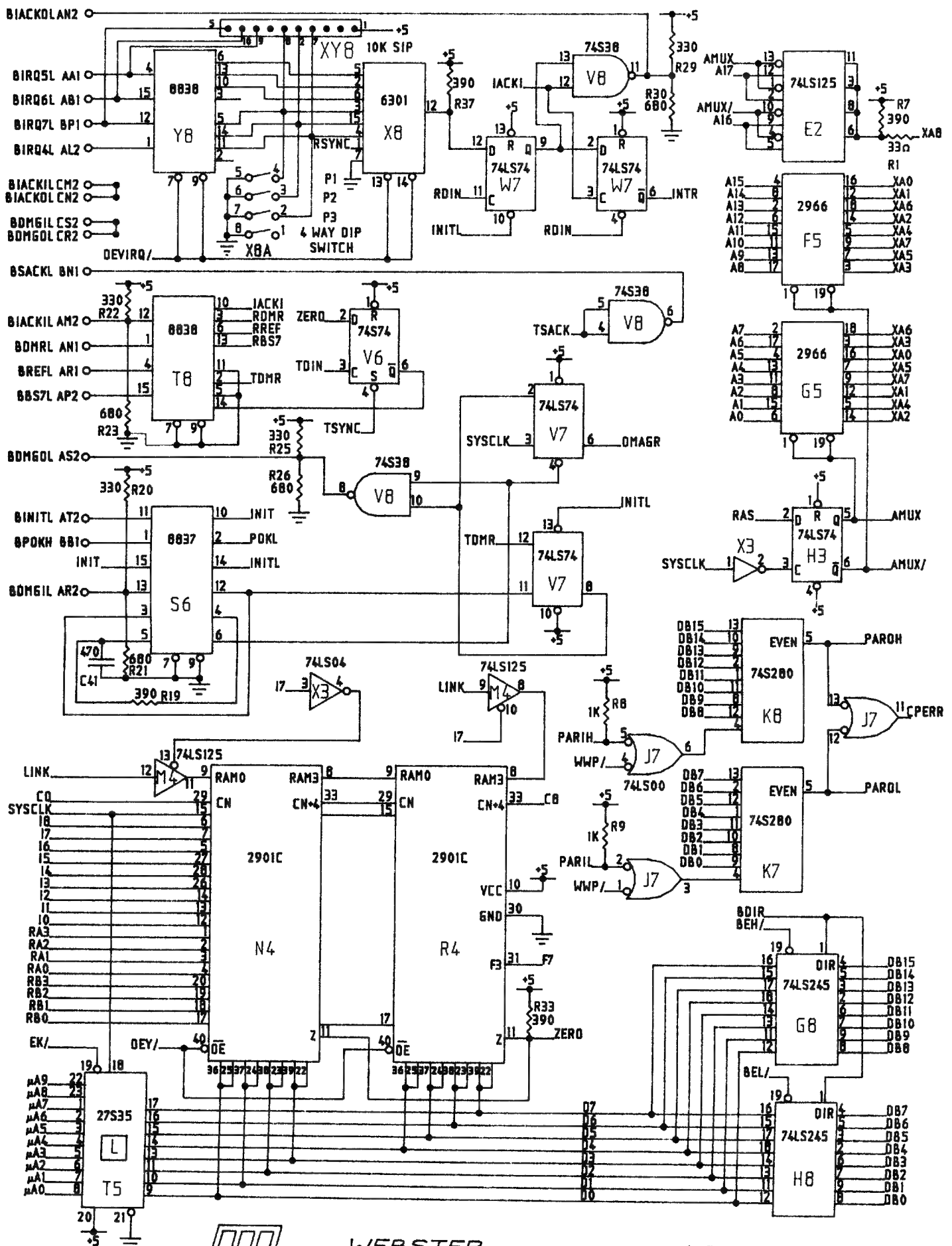
WEBSTER  
COMPUTER  
CORPORATION

WQESD  
DISC CONTROLLER  
SHEET 1 OF 7 - REVISED 25 SEP 90



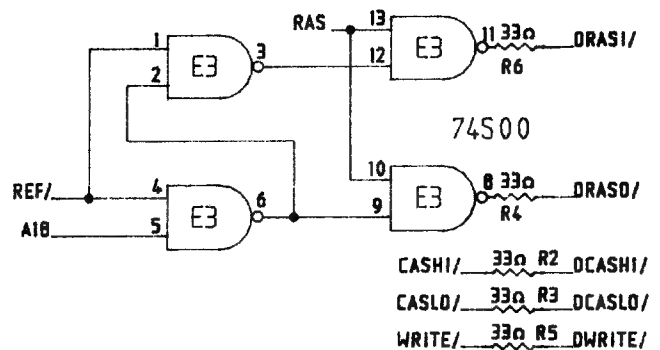
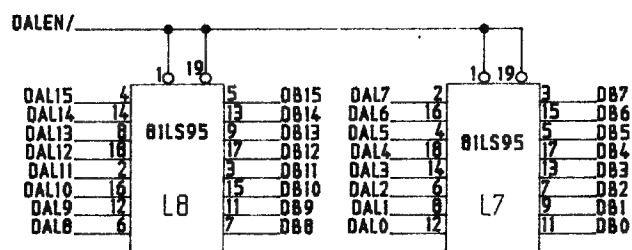
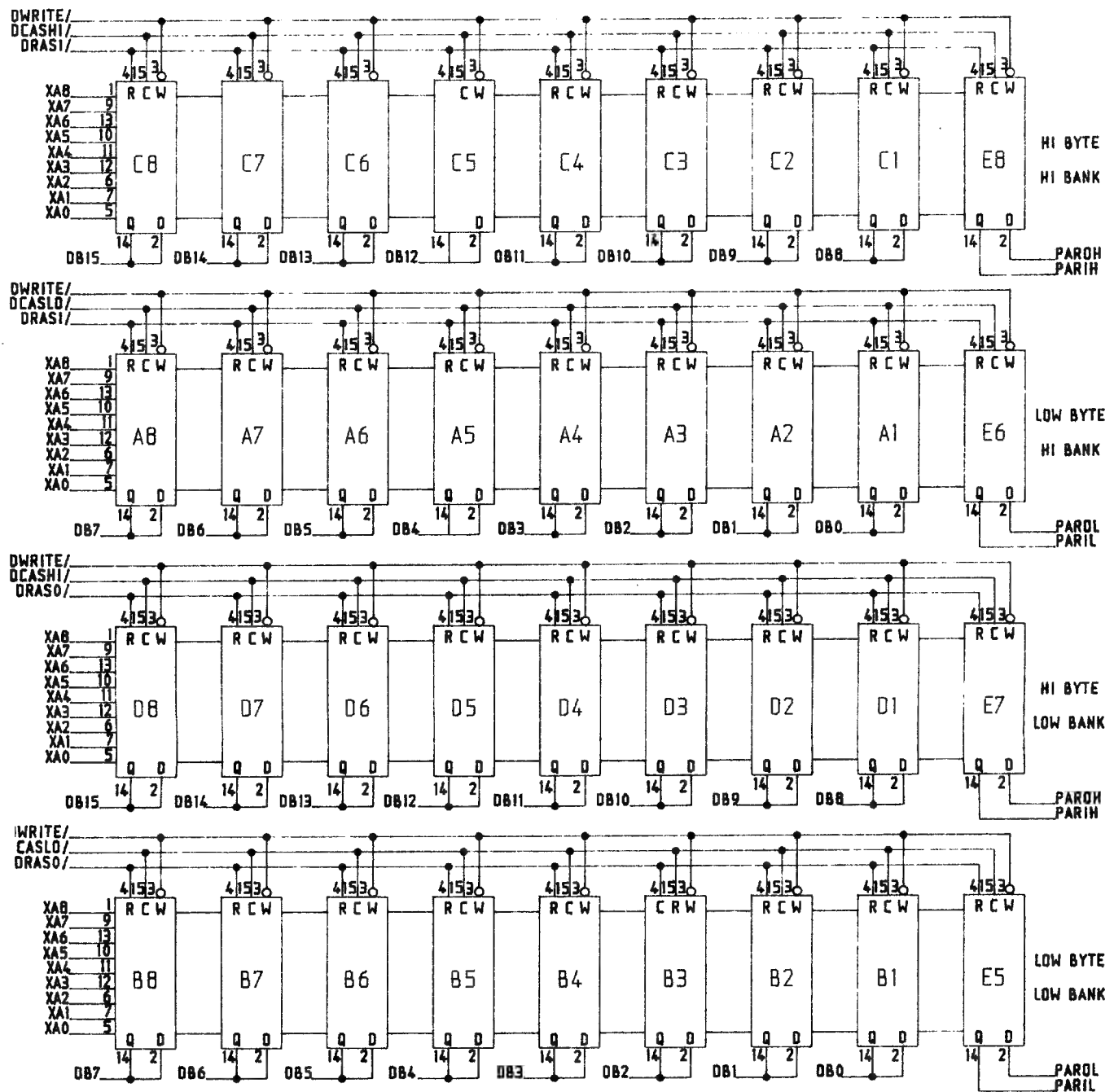
WEBSTER  
COMPUTER  
CORPORATION

WQESD  
DISC CONTROLLER  
SHEET 2 OF 7 - REVISED 25 SEP 90



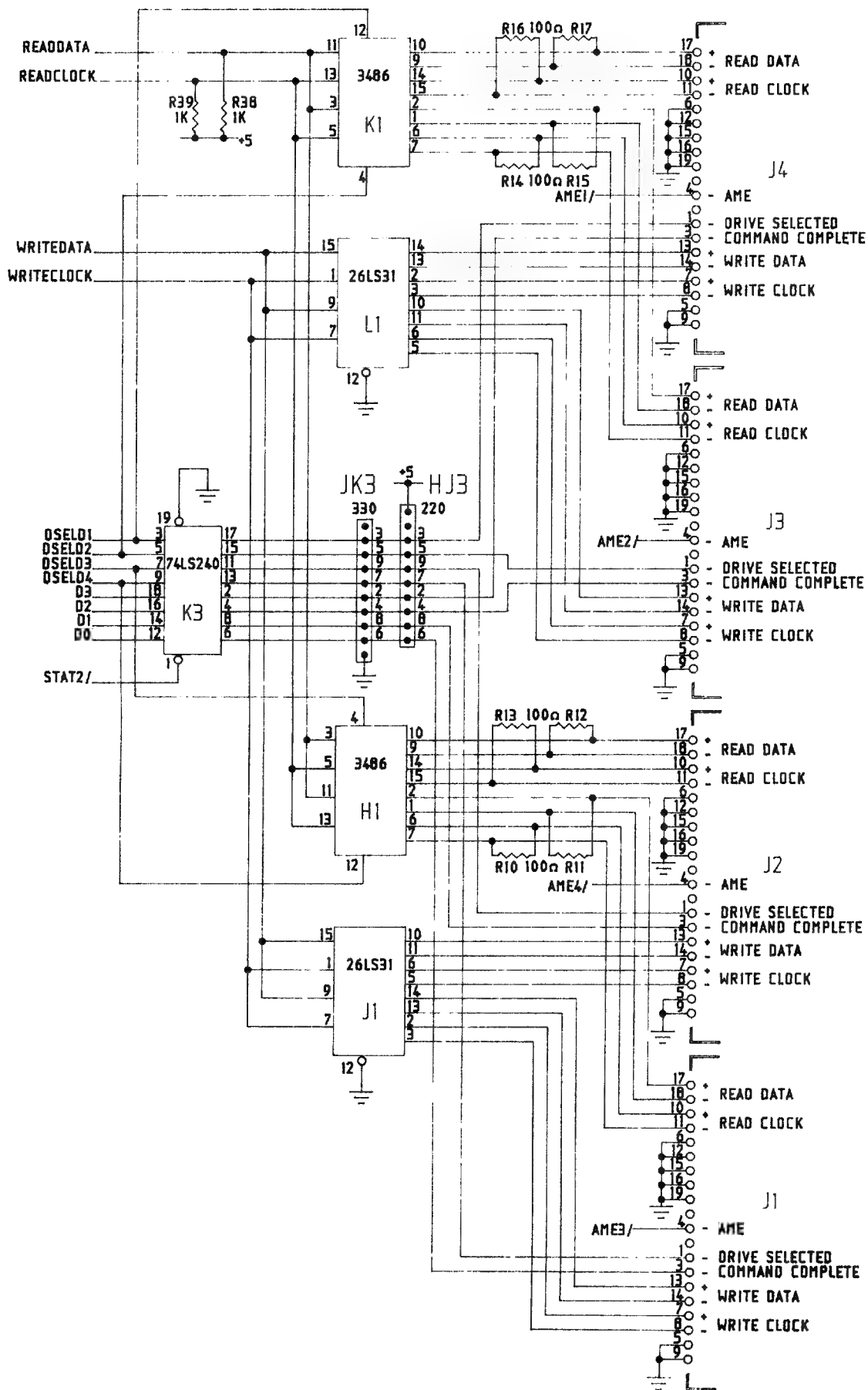
WEBSTER  
COMPUTER  
CORPORATION

WQESD  
DISC CONTROLLER  
SHEET 3 OF 7 - REVISED 25 SEP 90



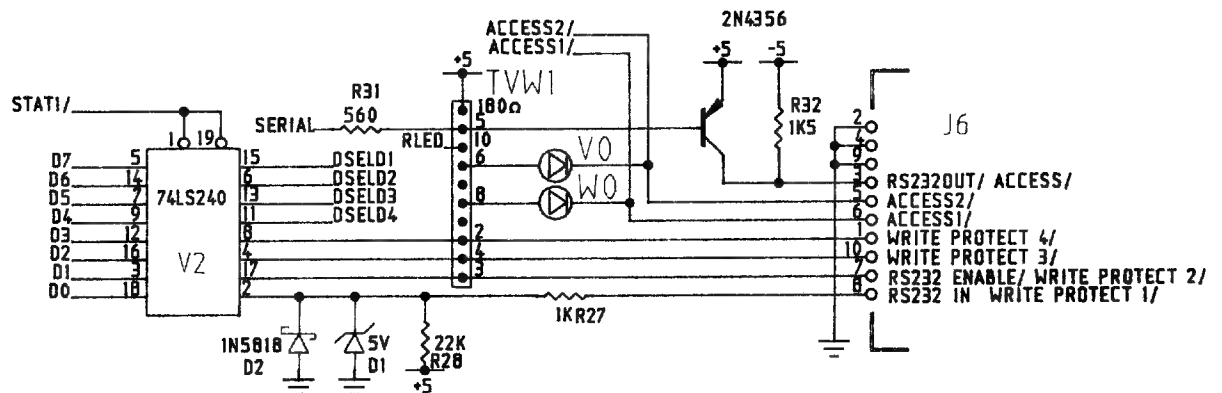
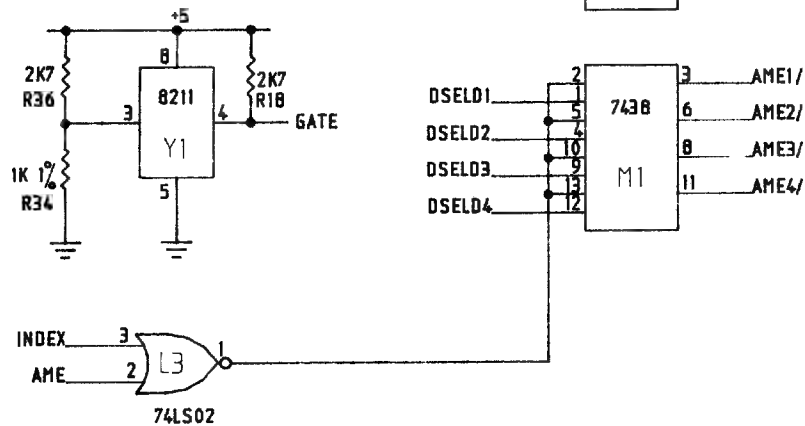
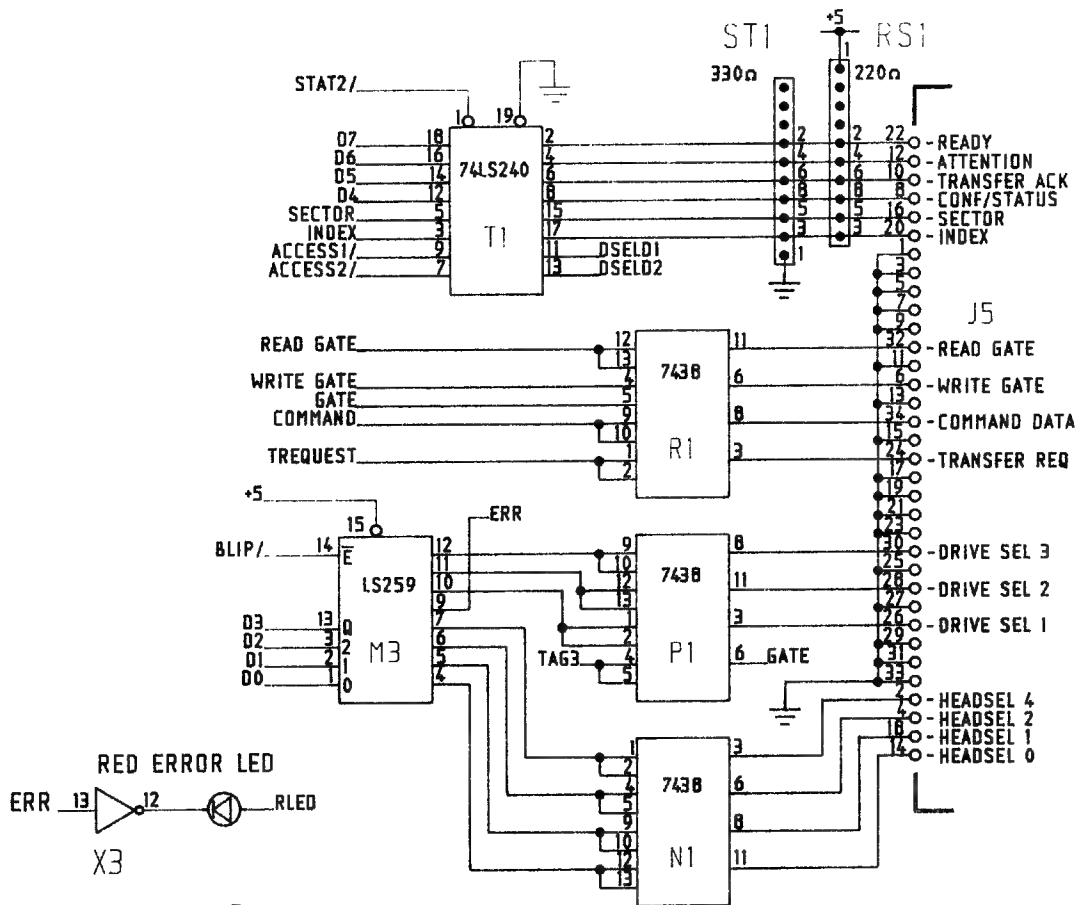
WEBSTER  
COMPUTER  
CORPORATION

QWESD  
DISC CONTROLLER  
SHEET 4 OF 7 - REVISED 25 SEP 80



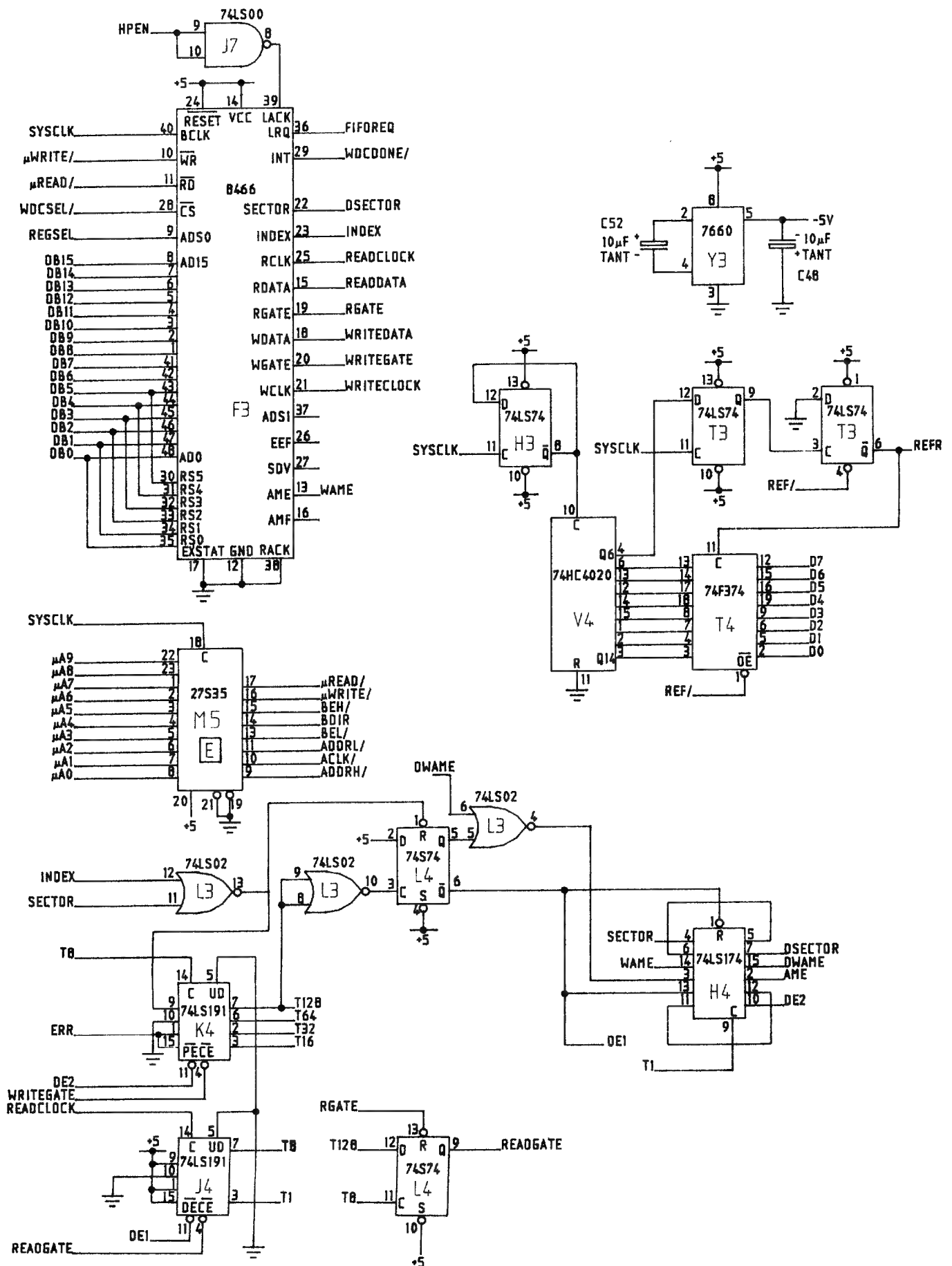
WEBSTER  
COMPUTER  
CORPORATION

WGSD  
DISC CONTROLLER  
SHEET 5 OF 7 - REVISED 25 SEP 90



WEBSTER  
COMPUTER  
CORPORATION

WQESD  
DISC CONTROLLER  
SHEET 6 OF 7 - REVISED 26-SEP-90



WEBSTER  
COMPUTER  
CORPORATION

WQESD  
DISC CONTROLLER

SHEET 7 OF 7 - REVISED 25 SEP 90

**Appendix A**  
**WQESD Sample Switch Settings**

**Maxtor EXT4000**

**1. Drive Configuration**

**a. Drive Select**

(DS1 - DS7). Looking at the drive PCB, there is a row of 7 jumpers near the 34 way edge connector. If no jumper is installed, the drive is not selected. The drive is factory configured as drive 1.

**b. Other Jumpers**

JP 6-8, JP 12-16 are for factory use, and should not need configuration. They should be set as follows:

JP 6 - JP 8 define drive type on the board.

	JP6	JP7	JP8
EXT-4125	open	short	open
EXT-4175	open	short	short
EXT-4280	short	open	short
EXT-4380	short	short	short

JP 12 - JP 16 are for factory testing and are NOT user settable. They should be set as follows:

JP 12	short
JP 13	short
JP 14	open
JP 15	open
JP 16	short



The other jumpers are for user options and should be set as follows:

JP 9        open    (Not write protected)  
JP 10       short (No Motor Control)  
JP 11       short (Off track error causes Write Fault)

The above selections for JP 9 to JP 11 are factory configurations.

## 2. WOMBAT Parameters

	EXT4175	EXT4280	EXT4380
Cylinders:	1223	1223	1223
Heads:	7	11	15
Sectors:	34	34	34
Track Spiralling:	4	4	4
Seek Optimisation:	1	1	1
Fairness Count:	24	24	24
Unit Number:	0	0	0
Unit Size:	size	size	size
Media Type:	DURA81	DURA81	DURA81
Unit Serial No.:	Drive S/N	Drive S/N	Drive S/N
Host Write Conf'n:	2	2	2
Unit Number:	(type RETURN)		
Read Lookahead:	4	4	4
Cache:	1	1	1
Command Queue Size:	8	8	8
Front Panel Type:	1	1	1

## Fujitsu M2246E

The parameters of the M224XC/E drives are determined by shorting links on the drive PCB.

The M224XC/E can be configured for either soft or hard sectoring. Hard sectoring is recommended.

For the M224XC/E rev A7 or earlier, with PCB B17B-0370-0020A rev 35G or earlier.

### 1. Drive Configuration

CNH1	No links installed	
CNH2	Link 15-16	
CNH3	No links installed	
CNH4	Link 3-4	
CNH5	Link 13-14	
CNH6	Link 1-2 (Drive 1, Daisy Chain)	(See *)
CNH7	Link 1-2, 5-6, 11-12. (No motor control, M2246E, Hard Sector, 32 Sectors/track)	(See %)

### \* Drive Address Configuration

Drive 1	Link 1-2 of CNH6
Drive 2	Link 3-4 of CNH6
Drive 3	Link 5-6 of CNH6
Drive 4	Link 7-8 of CNH6
Drive 5	Link 9-10 of CNH6
Drive 6	Link 11-12 of CNH6
Drive 7	Link 13-14 of CNH6

### % Drive Type Configuration

M2244C/E	Don't Link	5-6, 7-8	
M2245C/E	Don't Link	5-6, Link	7-8
M2246C/E	Don't Link	7-8, Link	5-6

## 2. WOMBAT Parameters

	M2244E	M2245E	M2246E
Cylinders:	822	822	822
Heads:	5	7	10
Sectors:	32	32	32
Track Spiralling:	4	4	4
Seek Optimisation:	1	1	1
Fairness Count:	24	24	24
Unit Number:	0	0	0
Unit Size:	size	size	size
Media Type:	DURA81	DURA81	DURA81
Unit Serial No.:	Drive S/N	Drive S/N	Drive S/N
Host Write Conf'n:	2	2	2
Unit Number:	(type RETURN)		
Read Lookahead:	4	4	4
Cache:	1	1	1
Command Queue Size:	8	8	8
Front Panel Type:	1	1	1

For the M224XC/E rev A8 or later, with PCB B17B-0370-0020A rev 36 H or later

### 1. Drive Configuration

CNH1	No links installed	
CNH2	Link 15-16	
CNH3	No links installed	
CNH4	Link 3-4	
CNH5	Link 13-14	
CNH6	Link 1-2 (Drive 1, Daisy Chain)	(See *)
CNH7	Link 1-2, 3-4, 7-8. (No motor control, M2246E, Hard Sector, 34 Sectors/track)	(See %)

### \* Drive Address Configuration

Drive 1	Link 1-2 of CNH6
Drive 2	Link 3-4 of CNH6
Drive 3	Link 5-6 of CNH6
Drive 4	Link 7-8 of CNH6
Drive 5	Link 9-10 of CNH6
Drive 6	Link 11-12 of CNH6
Drive 7	Link 13-14 of CNH6

### % Drive Type Configuration

M2244C/E	Don't Link	3-4, 5-6.
M2245C/E	Don't Link	3-4, Link 5-6.
M2246C/E	Don't Link	5-6, Link 3-4.

## 2. WOMBAT Parameters

	M2244E	M2245E	M2246E
Cylinders:	822	822	822
Heads:	5	7	10
Sectors:	34	34	34
Track Spiralling:	4	4	4
Seek Optimisation:	1	1	1
Fairness Count:	24	24	24
Unit Number:	0	0	0
Unit Size:	size	size	size
Media Type:	DURA81	DURA81	DURA81
Unit Serial No.:	Drive S/N	Drive S/N	Drive S/N
Host Write Conf'n:	2	2	2
Unit Number:	(type RETURN)		
Read Lookahead:	4	4	4
Cache:	1	1	1
Command Queue Size:	8	8	8
Front Panel Type:	1	1	1